L 12

```
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
SSS              YYY         YYY  SSS
SSS              YYY         YYY  SSS
SSS              YYY         YYY  SSS
SSS                 YYY   YYY     SSS
SSS                 YYY   YYY     SSS
SSS                 YYY   YYY     SSS
   SSSSSSSSS           YYY           SSSSSSSSS
   SSSSSSSSS           YYY           SSSSSSSSS
   SSSSSSSSS           YYY           SSSSSSSSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
```

B 13

```
**FILE**ID**FILEREAD





FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE  RRRRRRRR    EEEEEEEEEE    AAAAAA    DDDDDDDD
FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE  RRRRRRRR    EEEEEEEEEE    AAAAAA    DDDDDDDD
FF              II      LL          EE          RR      RR  EE          AA      AA  DD      DD
FF              II      LL          EE          RR      RR  EE          AA      AA  DD      DD
FF              II      LL          EE          RR      RR  EE          AA      AA  DD      DD
FF              II      LL          EE          RR      RR  EE          AA      AA  DD      DD
FFFFFFFF        II      LL          EEEEEEEE    RRRRRRRR    EEEEEEEE    AA      AA  DD      DD
FFFFFFFF        II      LL          EEEEEEEE    RRRRRRRR    EEEEEEEE    AA      AA  DD      DD
FF              II      LL          EE          RR  RR      EE          AAAAAAAAAA  DD      DD
FF              II      LL          EE          RR  RR      EE          AAAAAAAAAA  DD      DD
FF              II      LL          EE          RR    RR    EE          AA      AA  DD      DD      ....
FF              II      LL          EE          RR    RR    EE          AA      AA  DD      DD      ....
FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE  RR      RR  EEEEEEEEEE  AA      AA  DDDDDDDD        ....
FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE  RR      RR  EEEEEEEEEE  AA      AA  DDDDDDDD        ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
     (1)        78          DECLARATIONS
     (2)       179          FIL$OPENFILE - RETURN FILE HEADER AND STATISTICS BLOCK
     (3)       509          FIL$CACHE_INIT - INIT FILEREAD CACHE
     (4)       597          FIL$CACHE_TRUNC - TRUNCATE FILEREAD CACHE
     (5)       647          ASSIGN_DEV - ASSIGN A DEVICE AND RETURN A CHANNEL
     (6)       689          STORE3DIGITS - STORE 3 ASCII DIGITS
     (7)       724          FORMDIRSTRING - GET A DIRECTORY STRING
     (8)       791          MOUNT - MOUNT THE VOLUME, INIT FOR FILE LOOKUP
     (9)       899          FINDFILID - FIND FILE ID FOR SPECIFIED FILE
     (10)     1159          FIL$FINDFILID - STRUCTURE LEVEL 2
     (11)     1280          READ_DIR_LBN - READ NEXT DIRECTORY LBN
     (12)     1323          RDCHKFILHDR - READ AND CHECK FILE HEADER
     (13)     1468          READVBN, WRITEVBN - READ/WRITE VIRTUAL BLOCK
     (14)     1563          INIRTRVPTRSCAN - INITIALIZE RETRIEVAL POINTER SCAN
     (15)     1591          GETRTRVPTR - CONVERT NEXT RETRIEVAL POINTER
     (16)     1676          STATBLK - GET FILE STATISTICS BLOCK
     (17)     1790          FIL$CHKFILHDR - CHECK FILE HEADER VALIDITY
     (18)     1853          CHECKSUM - VALIDATE A CHECKSUM
```

```
0000        .TITLE  FILEREAD - FILES11 LEVEL 2 FILE READING ROUTINES
0000        .IDENT  'V04-000'
0000
0000   ;
0000   ;****************************************************************************
0000   ;*                                                                          *
0000   ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000   ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000   ;*  ALL RIGHTS RESERVED.                                                    *
0000   ;*                                                                          *
0000   ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000   ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000   ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000   ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000   ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000   ;*  TRANSFERRED.                                                            *
0000   ;*                                                                          *
0000   ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000   ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000   ;*  CORPORATION.                                                            *
0000   ;*                                                                          *
0000   ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000   ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000   ;*                                                                          *
0000   ;*                                                                          *
0000   ;****************************************************************************
0000   ;
0000   ;++
0000   ; FACILITY:     USER CALLABLE PROCEDURES
0000   ;
0000   ; ABSTRACT:
0000   ;
0000   ;       THIS SET OF ROUTINES PROVIDES THE CAPABILITY OF ''OPENING'' AND
0000   ; READING FILES BY FILE NAME FROM A FILES11 STRUCTURE LEVEL 2 VOLUME.
0000   ; THERE IS NO MULTI-VOLUME SUPPORT, AND MULTI-HEADER SUPPORT IS LIMITED
0000   ; TO RETURNING THE CORRECT FILE SIZE IN THE STATBLK.
0000   ;
0000   ; ENVIRONMENT:  USER MODE
0000   ;
0000   ; AUTHOR: PETER H. LIPMAN        , CREATION DATE: 14-DEC-76
0000   ;
0000   ; MODIFIED BY:
0000   ;
0000   ;       V03-010 WHM0006         Bill Matthews           21-Feb-1984
0000   ;               Removed all unneeded weak references.
0000   ;
0000   ;       V03-009 WHM0005         Bill Matthews           21-Feb-1984
0000   ;               Change sense of test for elements in search list from
0000   ;               length greater than zero to address not equal zero.
0000   ;
0000   ;       V03-008 WHM0004         Bill Matthews           16-Feb-1984
0000   ;               Continuation of changes of edit WHM0003.
0000   ;
0000   ;       V03-007 WHM0003         Bill Matthews           15-Feb-1984
0000   ;               Put new read/write storage declared in edit WHM0001 onto the
0000   ;               stack because there is no nice read/write psect for this data.
0000   ;
```

```
0000    ;       V03-006 WHM0002         Bill Matthews           13-Feb-1984
0000    ;               Bug fix to edit WHM0001.
0000    ;
0000    ;       V03-005 WHM0001         Bill Matthews           17-Jan-1984
0000    ;               Added support for a common system directory root.
0000    ;
0000    ;       V03-004 LJK0254         Lawrence J. Kenah       6-Dec-1983
0000    ;               Change LIB$CVT_DTB to FIL$CVT_DTB.
0000    ;
0000    ;       V03-003 RSH0052         R. Scott Hanna          22-Aug-1983
0000    ;               Add an optional input parameter that allows the caller to
0000    ;               modify the way the file open is performed. The caller
0000    ;               can specify that the cache is not to be used and/or that
0000    ;               the directory is not rooted.
0000    ;
0000    ;       V03-002 KDM0041         Kathleen D. Morse       14-Apr-1983
0000    ;               Remove the ODS-1 structure level support.
0000    ;
0000    ;--
```

```
                    0000                .SBTTL  DECLARATIONS
                    0000        ;
                    0000        ; INCLUDE FILES:
                    0000        ;
                    0000                .nocross
                    0000                $DIRDEF                         ;DIRECTORY ENTRY OFFSET DEFINITIONS
                    0000                $FATDEF                         ;RECORD ATTRIBUTE AREA DEFINITIONS
                    0000                $FH2DEF                         ;FILE HEADER DEFINITIONS, LEVEL 2
                    0000                $FM2DEF                         ;          MAP AREA, LEVEL 2
                    0000                $FIDDEF                         ;FILE ID OFFSET DEFINITIONS
                    0000                $HM2DEF                         ;HOME BLOCK DEFINITIONS, LEVEL 2
                    0000                $IODEF                          ;I/O DEFINITIONS
                    0000                $PSLDEF                         ;PROCESSOR STATUS LONG WORD DEFINITIONS
                    0000                $SSDEF                          ;SYSTEM SERVICE DEFINITIONS
                    0000        ;
                    0000        ; MACROS:
                    0000        ;
                    0000                .MACRO  READVBN CHAN,VBN,BUFADR,HDRADR
                    0000                .LIST   MEB
                    0000                        PUSHAL  HDRADR
                    0000                        PUSHAL  BUFADR
                    0000                        PUSHL   VBN
                    0000                        PUSHL   CHAN
                    0000                        CALLS   #4,W^FIL$READVBN
                    0000                .NLIST  MEB
                    0000                .ENDM   READVBN
                    0000
                    0000                .MACRO  READLBN CHAN,VBN,BUFADR
                    0000                .LIST   MEB
                    0000                        ROTL    #9,#1,-(SP)
                    0000                        MOVZWL  #IO$_READLBLK,-(SP)
                    0000                        PUSHAL  BUFADR
                    0000                        PUSHL   VBN
                    0000                        PUSHL   CHAN
                    0000                        CALLS   #5,W^FIL$RDWRTLBN
                    0000                .NLIST  MEB
                    0000                .ENDM   READLBN
                    0000                .cross
                    0000        ;
                    0000        ; EQUATED SYMBOLS:
                    0000        ;
                    0000
          000001FE  0000                FH2$W_VBNOFFSET = FH2$W_CHECKSUM ;SAVE INDEX FILE VBN OFFSET
                    0000                                                ;IN THIS PLACE IN INDEX FILE HEADER
                    0000                ASSUME  FH2$C_LEVEL2@-8 EQ 2
          00000009  0000                FH2$V_LEVEL2 = 9
          0000000A  0000                FH2$V_BIGFILNUM = 10            ;IF SET USE HIGH 8 BITS OF FILE ID RVN
                    0000                                                ;FIELD AS FILE NUMBER EXTENSION
                    0000                                                ;BIT IS PLACED IN FH2$W_STRUCLEV
                    0000                                                ;BY THE FIL$MOUNT CODE
                    0000
          00000001  0000                FIL$C_CACHE_ID = 1              ;VERSION OF THE FILEREAD CACHE
                    0000        ;
                    0000        ; OFFSETS INTO HEADER PORTION OF THE FILEREAD CACHE
                    0000        ;
                    0000                $OFFSET 0,POSITIVE,<-
                    0000                <FIL$W_CACHE_ID,2>,-            ;CACHE IDENT, AND WRITE INTERLOCK BIT
```

```
0000              <,2>,-                          ;SPARE
0000              FIL$L_DIROFF,-                  ;OFFSET IN BYTES TO DIRECTORY CACHE
0000              FIL$L_DIRNXT,-                  ;NEXT OFFSET TO ALLOCATE DIR CACHE ENTRY
0000              <FIL$L_DIRMAX,0>,-              ;MAX OFFSET FOR DIR CACHE
0000              FIL$L_LBNOFF,-                  ;OFFSET IN BYTES TO BEGIN OF LBN CACHE
0000              FIL$L_LBNNXT,-                  ;NEXT OFFSET TO ALLOCATE LBN CACHE
0000              FIL$L_LBNMAX,-                  ;MAX OFFSET FOR LBN CACHE
0000              <FIL$A_IXFHDR,512>,-            ;INDEX FILE HEADER
0000              <FIL$C_SIZE,0>-                 ;START OF DIRECTORY CACHE
0000              >
0000      FIL$W_CACHE_ID:
0004      FIL$L_DIROFF:
0008      FIL$L_DIRNXT:
000C      FIL$L_DIRMAX:
000C      FIL$L_LBNOFF:
0010      FIL$L_LBNNXT:
0014      FIL$L_LBNMAX:
0018      FIL$A_IXFHDR:
0218      FIL$C_SIZE:
0000      ;
0000      ; OFFSETS INTO DIRECTORY CACHE ENTRIES
0000      ;
0000              $OFFSET 0,POSITIVE,<-
0000              <FIL$A_DIR_FID,6>,-             ;DIRECTORY ID
0000              <FIL$T_DIR_NAM,10>,-            ;COUNTED NAME OF DIRECTORY - NO ".DIR"
0000              <FIL$Q_DIR_HDR,0>,-             ;DIRECTORY HEADER INFORMATION
0000              <FIL$W_DIR_BKCNT,2>,-           ;SIZE IN BLOCKS OF DIRECTORY FILE
0000              <FIL$B_DIR_LVL,1>,-             ;STRUCTURE LEVEL OF DIRECTORY
0000              <,1>,-                          ;SPARE BYTE
0000              FIL$L_DIR_LBN,-                 ;STARTING LBN OF DIRECTORY
0000              FIL$L_DIR_BFOFF,-               ;OFFSET TO DIR LBN BUFFER
0000              <FIL$W_DIR_BFCNT,2>,-           ;SIZE IN BLOCKS OF DIR LBN BUFFER
0000              <FIL$A_DIR_OFID,6>,-            ;OUTPUT FILE ID FROM LOOKUP
0000              <FIL$C_DIR_SIZE,0>-             ;SIZE OF DIRECTORY CACHE ENTRY
0000              >
0000      FIL$A_DIR_FID:
0006      FIL$T_DIR_NAM:
0010      FIL$Q_DIR_HDR:
0010      FIL$W_DIR_BKCNT:
0012      FIL$B_DIR_LVL:
0014      FIL$L_DIR_LBN:
0018      FIL$L_DIR_BFOFF:
001C      FIL$W_DIR_BFCNT:
001E      FIL$A_DIR_OFID:
0024      FIL$C_DIR_SIZE:
0000      ;
0000      ; MAKE THESE GLOBAL SO THAT A CACHE SIZE CAN BE PROPERLY CALCULATED
0000      ; THE CALCULATION IS:
0000      ;
0000      ;       FIL$C_SIZE + (DIRCNT * FIIL$C_DIR_SIZE> + <LBNCNT * 512>
0000      ;
0000              .GLOBAL FIL$C_SIZE,FIL$C_DIR_SIZE
0000
0000      ;
0000      ; READ-ONLY STORAGE:
0000      ;
0000
```

```
                                00000000   173            .PSECT  YFILEREAD,BYTE,EXE
                                    0000   174
                                    0000   175 CMNSYS:
     4E 4F 4D 4D 4F 43 53 59 53 00' 0000   176            .ASCIC  /SYSCOMMON/
                                09  0000
                                    000A   177
```

```
000A          .SBTTL  FIL$OPENFILE - RETURN FILE HEADER AND STATISTICS BLOCK
000A    ;++
000A    ; FUNCTIONAL DESCRIPTION:
000A    ;
000A    ;       THE OPENFILE ROUTINE ACCEPTS A FULL FILE NAME IN THE FORMAT
000A    ;       DEV:[DIR]FILE.TYP;VERSION.
000A    ; IT ASSIGNS AND RETURNS A CHANNEL, READS THE FILE HEADER, RETURNS THE
000A    ; STATISTICS BLOCK, AND OPTIONALLY RETURNS THE RETRIEVAL POINTERS IN
000A    ; A NORMALIZED (LONG WORD COUNT, LONG WORD LBN) FORMAT.
000A    ; THE DIRECTORY MAY BE IN ANY OF THE STANDARD FORMATS:
000A    ;       [10,40], [010040], [ABCDEFGHI], OR WITH < AND > REPLACING [ AND ].
000A    ; VERSION MAY BE ZERO IN WHICH CASE THE HIGHEST VERSION IS FOUND
000A    ;
000A    ; CALLING SEQUENCE:
000A    ;
000A    ;       CALLG   ARGLIST,FIL$OPENFILE
000A    ;
000A    ; INPUT PARAMETERS:
000A    ;
000A    ;       CHANADR(AP)     =       ADDRESS TO RETURN CHANNEL
000A    ;       FILNAM(AP)      =       ADDRESS OF 2 LONG WORD FILE NAME STRING DESCRIPTOR
000A    ;                               1 - SIZE OF STRING
000A    ;                               2 - ADDRESS OF STRING
000A    ;                                       DB1:[10,40]FILTST.EXE
000A    ;       IXFHDR(AP)      =       ADDRESS OF 512 BYTE BUFFER TO BE USED FOR
000A    ;                                       THE INDEX FILE HEADER
000A    ;       FILHDR(AP)      =       ADDRESS OF 512 BYTE BUFFER TO RETURN FILE HEADER
000A    ;       STATBLK(AP)     =       ADDRESS OF 2 LONG WORD BLOCK IN WHICH THE
000A    ;                               FOLLOWING WILL BE RETURNED
000A    ;                               1 - LOGICAL BLOCK NUMBER OF FIRST BLOCK OF
000A    ;                                       FILE OR 0 IF FILE IS NOT CONTIGUOUS
000A    ;                               2 - SIZE OF FILE IN BLOCKS
000A    ;       RTRVPTRLEN(AP)  =       ADDRESS TO RETURN THE NUMBER OF
000A    ;                               BYTES OF RETRIEVAL POINTERS STORED
000A    ;                               THIS PARAMETER IS OPTIONAL. IT IS NOT USED
000A    ;                               IF MISSING OR SPECIFIED AND 0.
000A    ;       RTRVPTRBUF(AP)  =       ADDRESS OF RETRIEVAL POINTER BUFFER DESCRIPTOR.
000A    ;                               THIS PARAMETER SHOULD BE SPECIFIED IF AND ONLY
000A    ;                               IF RTRVPTRLEN IS PRESENT.
000A    ;                               THE RETRIEVAL POINTERS ARE RETURNED IN
000A    ;                               THE FORM 32 BIT BLOCK COUNT, 32 BIT LBN
000A    ;                               A ZERO BUFFER DESCRIPTOR ADDRESS OR A
000A    ;                               ZERO BUFFER ADDRESS MEANS DON'T
000A    ;                               RETURN RETRIEVAL POINTER INFO
000A    ;       FLAGS(AP)       =       FLAGS LONGWORD. THE FOLLOWING BITS MODIFY
000A    ;                               THE WAY THE FILE OPEN IS PERFORMED.
000A    ;
000A    ;                               BIT 0   WHEN SET THE CACHE IS NOT USED.
000A    ;
000A    ;                               BIT 1   WHEN SET THE DIRECTORY IS NOT ASSUMED
000A    ;                                       TO BE ROOTED.
000A    ;
000A    ; IMPLICIT INPUTS:
000A    ;
000A    ;       NONE
000A    ;
000A    ; OUTPUT PARAMETERS:
```

```
                  000A   236 ;
                  000A   237 ;          R0 = SYSTEM STATUS CODE
                  000A   238 ;
                  000A   239 ; IMPLICIT OUTPUTS:
                  000A   240 ;
                  000A   241 ;          NONE
                  000A   242 ;
                  000A   243 ; COMPLETION CODES:
                  000A   244 ;
                  000A   245 ;          SS$_NORMAL                  SUCCESSFUL COMPLETION
                  000A   246 ;          SS$_NOSUCHFILE              FAILED TO FIND DIRECTORY OR FILE
                  000A   247 ;          SS$_BADFILENAME             SYNTAX ERROR IN DIRECTORY OR FILE NAME STRING
                  000A   248 ;
                  000A   249 ; THE FOLLOWING COMPLETION CODES INDICATE FILE STRUCTURE PROBLEMS
                  000A   250 ;
                  000A   251 ;          SS$_BADCHKSUM               CHECKSUM ERROR IN HOME BLOCK, INDEX FILE HEADER
                  000A   252 ;                                      DIRECTORY FILE HEADER OR FILE HEADER
                  000A   253 ;          SS$_BADFILEHDR              FILE HEADER CONSISTENCY CHECK FAILED FOR
                  000A   254 ;                                      INDEX FILE, DIRECTORY FILE, OR DESIRED FILE
                  000A   255 ;          SS$_FILESTRUCT              HOME BLOCK INDICATES THAT THIS VOLUME
                  000A   256 ;                                      CONTAINS A NON-SUPPORTED FILE STRUCTURE
                  000A   257 ;                                      OR POSSIBLY THE HOMEBLOCK IS GARBAGE
                  000A   258 ;
                  000A   259 ; SIDE EFFECTS:
                  000A   260 ;
                  000A   261 ;          NONE
                  000A   262 ;
                  000A   263 ; EQUATED SYMBOLS
                  000A   264 ;
                  000A   265 ;      OFFSETS FROM AP
                  000A   266 ;
        00000000  000A   267            ARGCNT          = 0
        00000004  000A   268            CHANADR         = 4
        00000008  000A   269            FILNAM          = 8
        0000000C  000A   270            IXFHDR          = 12
        00000010  000A   271            FILHDR          = 16
        00000014  000A   272            STATBLK         = 20
        00000018  000A   273            RTRVPTRLEN      = 24                    ; OPTIONAL
        0000001C  000A   274            RTRVPTRBUF      = 28                    ; OPTIONAL
        00000020  000A   275            FLAGS           = 32                    ; OPTIONAL
                  000A   276
                  000A   277 ;
                  000A   278 ; SIZE OF DIRECTORY SEARCH LIST
                  000A   279 ;
        00000018  000A   280            DIR_LIST_SIZE   = 24
                  000A   281
                  000A   282 ;
                  000A   283 ;      OFFSETS FROM FP
                  000A   284 ;
                  000A   285            $OFFSET 0,NEGATIVE,<-
                  000A   286            <FID,6>,-                           ;3 WORD FILE IDENTIFIER
                  000A   287            <DIRNAM,16>,-                       ;DIRECTORY NAME AREA
                  000A   288            <NAMBLK,10>,-                       ;5 WORD NAME BLOCK AREA
                  000A   289            <NAMDSC,12>,-                       ;NAME DESCRIPTOR AREA
                  000A   290            <LOCAL_FLAGS,4>,-                   ;LOCAL COPY OF FLAGS INPUT PARAMETER
                  000A   291            <DIR_DESC,8>,-                      ;RW STORAGE TO HOLD THE DIR STRING
                  000A   292            <FILE_NAME_DESC,8>,-                ;RW STORAGE TO HOLD THE FILENAME STRING
```

```
000A   293          <PREFIX_DIR_LIST,DIR_LIST_SIZE>,-;RW STORAGE TO HOLD LIST OF DIRECTORIES
000A   294          <SCRATCHSIZE,0>-                ;SIZE OF SCRATCH AREA
000A   295          >
FFFA          FID:
FFEA          DIRNAM:
FFE0          NAMBLK:
FFD4          NAMDSC:
FFD0          LOCAL_FLAGS:
FFC8          DIR_DESC:
FFC0          FILE_NAME_DESC:
FFA8          PREFIX_DIR_LIST:
FFA8          SCRATCHSIZE:
000A   296    ;
000A   297    ; THE LIST OF DIRECTORIES TO BE SEARCHED LOOKS AS FOLLOWS
000A   298    ;
000A   299    ; PREFIX_DIR_LIST:
000A   300    ;        +----------------------------+
000A   301    ;        ! PTR TO TOP LEVEL SYS NAME !\
000A   302    ;        !----------------------------! ! SEARCH FOR SYSN.XXXXXX
000A   303    ;        !           ZERO             !/
000A   304    ;        +----------------------------+
000A   305    ;        ! PTR TO TOP LEVEL SYS NAME !\
000A   306    ;        +----------------------------+ !
000A   307    ;        ! PTR TO SYSCOMMON NAME STR ! !
000A   308    ;        +----------------------------+ ! SEARCH FOR SYSN.SYSCOMMON.XXXXXX
000A   309    ;        !           ZERO             ! !
000A   310    ;        +----------------------------+/
000A   311    ;        !           ZERO             !   END OF DIRECTORY SEARCH LIST
000A   312    ;        +----------------------------+
000A   313
000A   314
000A   315    ; THE FILE DESCRIPTION ON THE STACK LOOKS AS FOLLOWS:
000A   316    ;
000A   317    ;        +----------------------------+
000A   318    ;        !  DIRECTORY NAME COUNT      !   :NAMDSC
000A   319    ;        !----------------------------!
000A   320    ;        ! ADDRESS OF DIRECTORY NAME  !
000A   321    ;        +----------------------------+
000A   322    ;        !    ADDRESS OF NAMBLK       !
000A   323    ;        +----------------------------+
000A   324    ;        !                            !   :NAMBLK
000A   325    ;        +----------------------------+
000A   326    ;        !                            !
000A   327    ;        +----------------------------+
000A   328    ;        !                            !
000A   329    ;        !                            !
000A   330    ;        +----------------------------+
000A   331    ;        !                            !   :DIRNAM
000A   332    ;        !      ASCII STRING OF       !
000A   333    ;        !      DIRECTORY FILE        !
000A   334    ;        !    NAME.TYPE;VERSION       !
000A   335    ;        !                            !
000A   336    ;        +----------------------------+
000A   337    ;        !          FILE ID           !   :FILID
000A   338    ;        !          NUMBER            !
000A   339    ;        +----------------------------+
000A   340    ;        !  RELATIVE VOLUME NUMBER    !
```

```
                              000A   341 ;          +---------------------------+
                              000A   342 ;
                              000A   343 ;--
                              000A   344
                              000A   345            .ENABL  LSB
                              000A   346
                              000A   347 FIL$OPENFILE::
                        09FC  000A   348            .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R11>
               0204   30  000C   349            BSBW    ASSIGN_DEV                ;ASSIGN THE DEVICE ONCE IN CALLER'S MODE
              03 50   E8  000F   350            BLBS    R0,OPENFILE_2             ;BRANCH IF SUCCESSFUL
                      04  0012   351            RET
                          0013   352 FIL$OPENFILE_1::
                    09FC  0013   353            .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R11>
                          0015   354 OPENFILE_2:
 5E    00000058 8F    C2  0015   355            SUBL    #-SCRATCHSIZE,SP          ;RESERVE SCRATCH STORAGE
    DC AD    E0 AD    DE  001C   356            MOVAL   NAMBLK(FP),NAMDSC+8(FP) ;SET ADDRESS OF NAME BLOCK
             D0 AD    D4  0021   357            CLRL    LOCAL_FLAGS(FP)           ;ASSUME NO FLAGS PARAMETER
18    00    6E    00  2C  0024   358            MOVC5   #0,(SP),#0,#DIR_LIST_SIZE,-;CLEAR DIRECTORY SEARCH LIST BLOCK
             A8 AD        0029   359                    PREFIX_DIR_LIST(FP)       ;
          08    6C    D1  002B   360            CMPL    ARGCNT(AP),#FLAGS/4       ;IS THERE A FLAGS PARAMETER?
                09    19  002E   361            BLSS    1$                        ;BR IF NOT
    D0 AD    20 AC    D0  0030   362            MOVL    FLAGS(AP),LOCAL_FLAGS(FP) ;GET COPY OF FLAGS
          34 D0 AD    E8  0035   363            BLBS    LOCAL_FLAGS(FP),15$      ;BR IF WE ARE NOT TO USE THE CACHE
                          0039   364 ;
                          0039   365 ; IF CACHE DESCRIPTOR EXISTS AND IS IN SYSTEM SPACE, THEN WE
                          0039   366 ; HAD BETTER BE IN KERNEL MODE TO USE THE CACHE.
                          0039   367 ;
 5B    00000000'EF    DE  0039   368 1$:        MOVAL   FIL$GQ_CACHE,R11          ;IS CACHE IN SYSTEM SPACE?
                1D    14  0040   369            BGTR    10$                       ;BRANCH IF DESCRIPTOR PRESENT
                          0042   370                                              ;AND NOT IN SYSTEM SPACE
                2B    13  0042   371            BEQL    20$                       ;BRANCH IF NO DESCRIPTOR PRESENT
                50    DC  0044   372            MOVPSL  R0                        ;GET PSL
50    50    02    18  EF  0046   373            EXTZV   #PSL$V_CURMOD,#PSL$S_CURMOD,R0,R0 ;FETCH CURRENT MODE
                12    13  004B   374            BEQL    10$                       ;BRANCH IF ALREADY IN KERNEL MODE
                          004D   375            $CMKRNL_S B^FIL$OPENFILE_1,(AP) ;CALL THIS PROCEDURE IN KERNEL MODE
          24    50    D1  0059   376            CMPL    R0,#SS$_NOPRIV            ;ASSUME NOPRIV MEANS WE COULDN'T
                          005C   377                                              ;GET IN TO KERNEL MODE
                0F    13  005C   378            BEQL    15$
                      04  005E   379            RET
 5B    00000004'EF    D0  005F   380 10$:       MOVL    FIL$GQ_CACHE+4,R11        ;IS THE CACHE ENABLED?
                07    13  0066   381            BEQL    20$                       ;BRANCH IF NOT
          01    6B    B1  0068   382            CMPW    FIL$W_CACHE_ID(R11),#FIL$C_CACHE_ID ;CORRECT VERSION OF CACHE?
                02    13  006B   383            BEQL    20$                       ;BRANCH IF YES
                5B    D4  006D   384 15$:       CLRL    R11                       ;DISABLE THE CACHE
 57    00000000'EF    DE  006F   385 20$:       MOVAL   FIL$GT_DDSTRING,R7        ;ADDRESS OF COUNTED STRING
          56    87    9A  0076   386            MOVZBL  (R7)+,R6                  ;GET BYTE COUNT
                57    D6  0079   387            INCL    R7                        ;STEP OVER BRACKET
          56    02    C2  007B   388            SUBL    #2,R6                     ;DON'T COUNT THE BRACKETS
                          007E   389 ;
                          007E   390 ; GET FILE NAME STRING, AND STRIP DEVICE OFF IF PRESENT
                          007E   391 ;
       50    08 AC    D0  007E   392            MOVL    FILNAM(AP),R0             ;ADDRESS OF FILE NAME DESCRIPTOR
                27    13  0082   393            BEQL    32$                       ;BRANCH IF NO NAME SPECIFIED
          52    60    7D  0084   394            MOVQ    (R0),R2                   ;R2 = SIZE, R3 = ADDRESS
    63    52    3A    3A  0087   395            LOCC    #^A/:/,R2,(R3)            ;DEVICE NAME PRESENT?
                07    13  008B   396            BEQL    25$                       ;BRANCH IF NOT
       53    01 A1    9E  008D   397            MOVAB   1(R1),R3                  ;ADDRESS BEYOND ":"
```

```
        52   70   9E  0091            MOVAB   -(R0),R2                       ;REMAINING SIZE
                      0094       ;
                      0094       ; SEE IF DIRECTORY SPECIFIED IN THE FILE NAME STRING
                      0094       ;
    63   5B 8F    91  0094       25$:    CMPB    #^A/[/,(R3)                    ;DIRECTORY DELIMITER?
              05  13  0098            BEQL    30$                            ;BRANCH IF YES
        63   3C   91  009A            CMPB    #^A/</,(R3)                    ;ALTERNATE CHARACTER
              1D  12  009D            BNEQ    40$                            ;BRANCH IF NO DIRECTORY SPECIFIED
  50    83   02   81  009F       30$:    ADDB3   #2,(R3)+,R0                    ;SCAN FOR MATCHING BRACKET ] OR >
              52  D7  00A3            DECL    R2                             ;ADJUST SIZE AND ADR OF STRING
  63    52   50   3A  00A5            LOCC    R0,R2,(R3)                     ;SCAN FOR CLOSE BRACKET
              03  12  00A9            BNEQ    35$                            ;BRANCH IF FOUND IT
          03FC    31  00AB       32$:    BRW     BADFILNAM                      ;BAD FILE NAME IF NO CLOSE BRACKET
        57   53   D0  00AE       35$:    MOVL    R3,R7                          ;ADDRESS OF DIRECTORY NAME
  56    51   53   C3  00B1            SUBL3   R3,R1,R6                       ;SIZE OF DIRECTORY NAME
        52   70   9E  00B5            MOVAB   -(R0),R2                       ;SIZE REMAINING SKIP CLOSE BRACKET
    53   01 A1    DE  00B8            MOVAL   1(R1),R3                       ;ADR OF REMAINING STRING BEYOND CLOSE BRACKE
                      00BC       ;
                      00BC       ; SET UP COMMON ARGUMENT LIST FOR MOUNT, FINDFILID, RDCHKFILHDR
                      00BC       ;
                      00BC       ;       +-----------------------------+
                      00BC       ;       !       ARGUMENT COUNT        !   : AP
                      00BC       ;       +-----------------------------+
                      00BC       ;       !       CHANNEL NUMBER        !
                      00BC       ;       +-----------------------------+
                      00BC       ;       !       NAME DESCRIPTOR       !
                      00BC       ;       +-----------------------------+
                      00BC       ;       ! INDEX FILE HEADER BUF ADR   !
                      00BC       ;       +-----------------------------+
                      00BC       ;       !  FILE HEADER BUFFER ADDR    !
                      00BC       ;       +-----------------------------+
                      00BC       ;       ! ADDR OF STATISTICS BLOCK    !
                      00BC       ;       +-----------------------------+
                      00BC       ;       ! ADDRESS OF FILE ID BLOCK    !
                      00BC       ;       +-----------------------------+
                      00BC       ;       !  ADDR OF RTRV PTR LENGTH    !
                      00BC       ;       +-----------------------------+
                      00BC       ;       ! ADDR OF RTRV PTR BUF DSCR   !
                      00BC       ;       +-----------------------------+
                      00BC       ;
             7E   7C  00BC       40$:    CLRQ    -(SP)                          ;ASSUME NO RETRIEVAL POINTERS REQUESTED
        07   6C   D1  00BE            CMPL    ARGCNT(AP),#RTRVPTRBUF/4       ;RETRIEVAL POINTER PARAMETERS PRESENT?
             04   19  00C1            BLSS    45$                            ;BRANCH IF NOT
    6E   18 AC    7D  00C3            MOVQ    RTRVPTRLEN(AP),(SP)            ;PUT RTRV PTR PARAMS IN LIST
         FA AD    DF  00C7       45$:    PUSHAL  FID(FP)                        ;ADDRESS OF FILE ID
    7E   10 AC    7D  00CA            MOVQ    FILHDR(AP),-(SP)               ;PUSH STATBLK ADR, FILHDR ADR
         0C AC    DD  00CE            PUSHL   IXFHDR(AP)                     ;INDEX FILE HEADER ADDRESS
         D4 AD    DF  00D1            PUSHAL  NAMDSC(FP)                     ;ADR OF 3 LONG WORD NAME DESCRIPTOR
         04 BC    DD  00D4            PUSHL   @CHANADR(AP)                   ;CHANNEL TO USE, LONG WORD FOR BOOTING
             06   DD  00D7            PUSHL   #6                             ;PARAMETER COUNT
             5B   D5  00D9            TSTL    R11                            ;CACHE ENABLED?
             07   13  00DB            BEQL    50$
 0C AE   18 AB    DE  00DD            MOVAL   FIL$A_IXFHDR(R11),IXFHDR(SP)   ;USE CACHED INDEX FILE HEADER
             0B   11  00E2            BRB     60$                            ;AND SKIP THE MOUNT
 02D3'CF     6E   FA  00E4       50$:    CALLG   (SP),W^FIL$MOUNT               ;"MOUNT THE VOLUME" (READ HOME
                      00E9                                                   ;BLOCK, INDEX FILE HEADER, GET
                      00E9                                                   ;STRUCTURE LEVEL OF VOLUME)
             03 50   E8  00E9    455           BLBS    R0,60$                     ;BRANCH IF SUCCESS
              0090   31  00EC    456           BRW     100$                       ;ELSE RETURN WITH ERROR STATUS
                         00EF    457 ;
                         00EF    458 ; SET UP FOR THE DIRECTORY LOOK UP
                         00EF    459 ;
      C0 AD     52   7D  00EF    460 60$:      MOVQ    R2,FILE_NAME_DESC(FP)      ;SAVE A DESC OF THE FILENAME STRING
      C8 AD     56   7D  00F3    461           MOVQ    R6,DIR_DESC(FP)            ;SAVE THE DIRECTORY DESCRIPTOR
      58     A8 AD   DE  00F7    462           MOVAL   PREFIX_DIR_LIST(FP),R8     ;GET PTR TO PREFIX DIRECTORY SEARCH LIST
   11 D0 AD     01   E0  00FB    463           BBS     #1,LOCAL_FLAGS(FP),65$     ;BR IF DIRECTORY ID NOT ROOTED
68   00000000'EF     DE  0100    464           MOVAL   FIL$GT_TOPSYS,(R8)         ;GET PTR TO TOP LEVEL SYS NAME STRING
         08 A8  68   D0  0107    465           MOVL    (R8),8(R8)                 ;SET TOP LEVEL SYS NAME IN SYSCOMMON LIST
  0C A8     FEF1 CF  DE  010B    466           MOVAL   CMNSYS,12(R8)              ;SET SYSCOMMON NAME ADDRESS IN SYSCOMMON LIS
      FA AD     04   B0  0111    467 65$:      MOVW    #FID$C_MFD,FID(FP)         ;MFD FILE NUMBER
      FC AD     04   D0  0115    468           MOVL    #FID$C_MFD,FID+2(FP)       ;MFD FILE SEQUENCE NO., RVN = 0
         51     88   D0  0119    469 67$:      MOVL    (R8)+,R1                   ;GET PTR TO A DIRECTORY NAME STRING
                11   13  011C    470           BEQL    70$                        ;IF EQL NONE SPECIFIED
         50     81   9A  011E    471           MOVZBL  (R1)+,R0                   ;GET SIZE TO R0, ADR TO R1
                0C   13  0121    472           BEQL    70$                        ;BRANCH IF NONE SPECIFIED
         56     50   7D  0123    473           MOVQ    R0,R6                      ;TREAT TOPSYS OR CMNSYS LIKE DIR STRING
              013E   30  0126    474           BSBW    FORMDIRSTRING              ;FORM THE DIRECTORY NAME
      56     C8 AD   7D  0129    475           MOVQ    DIR_DESC(FP),R6            ;RESTORE READ DIRECTORY DESCRIPTOR
                03   11  012D    476           BRB     75$
              0135   30  012F    477 70$:      BSBW    FORMDIRSTRING              ;GET NEXT DIRECTORY TO LOOKUP
      D4 AD     50   7D  0132    478 75$:      MOVQ    R0,NAMDSC(FP)              ;STORE DESCRIPTOR OF ITS NAME
                6E   DF  0136    479 80$:      PUSHAL  (SP)                       ;REAL ADDRESS OF ARGUMENT LIST
                5B   DD  0138    480           PUSHL   R11                        ;CACHE ADDRESS IF ANY
      0358'CF   02   FB  013A    481           CALLS   #2,W^FIL$FINDFILID         ;FIND THE FILE ID
             17 50   E8  013F    482           BLBS    R0,83$                     ;BRANCH IF SUCCESS
 00000910 8F    50   D1  0142    483           CMPL    R0,#SS$_NOSUCHFILE         ;COULDN'T FIND THE FILE?
                34   12  0149    484           BNEQ    100$                       ;IF NEQ NO THEN RETURN ERROR STATUS
                88   D5  014B    485           TSTL    (R8)+                      ;POP OFF END OF DIR NAME STRING LIST MARKER
                30   12  014D    486           BNEQ    100$                       ;IF NEQ THEN ERROR
                68   D5  014F    487           TSTL    (R8)                       ;NOW AT END OF LIST?
                2C   13  0151    488           BEQL    100$                       ;IF EQL YES REPORT ERROR
      52     C0 AD   7D  0153    489           MOVQ    FILE_NAME_DESC(FP),R2      ;RESTORE THE FILE NAME STRING DESCRIPTOR
                B8   11  0157    490           BRB     65$                        ;TRY AGAIN USING COMMON SYS DIRECTORY
             E0 AD   7C  0159    491 83$:      CLRQ    NAMBLK(FP)                 ;REINIT NAME BLOCK
             E8 AD   B4  015C    492           CLRW    NAMBLK+8(FP)
                68   D5  015F    493           TSTL    (R8)                       ;ANY MORE PREFIX DIRECTORY NAMES?
                B6   12  0161    494           BNEQ    67$                        ;IF NEQ YES
                56   D5  0163    495           TSTL    R6                         ;ANY MORE DIRECTORY NAMES?
                C8   14  0165    496           BGTR    70$                        ;BRANCH IF YES, LOOKUP THE NEXT
      D4 AD     52   7D  0167    497           MOVQ    R2,NAMDSC(FP)              ;DESCRIPTOR FOR FILE TO LOOKUP
                52   D4  016B    498           CLRL    R2                         ;STOP THE LOOKUP LOOP
             D4 AD   D5  016D    499           TSTL    NAMDSC(FP)                 ;ALREADY DONE?
                C4   14  0170    500           BGTR    80$                        ;BRANCH IF NO, DO THE LAST ONE
         07     6C   D1  0172    501 85$:      CMPL    ARGCNT(AP),#RTRVPTRBUF/4   ;RETRIEVAL POINTERS DESIRED?
                03   19  0175    502           BLSS    90$                        ;BRANCH IF NOT
         6E     02   C0  0177    503           ADDL    #2,(SP)                    ;ADDITIONAL ARGUMENTS ARE PRESENT
      05BA'CF   6E   FA  017A    504 90$:      CALLG   (SP),W^FIL$RDCHKFILHDR     ;READ AND CHECK FILE HEADER
                     04  017F    505 100$:     RET
                         0180    506
                         0180    507           .DSABL  LSB
```

```
               0180          .SBTTL  FIL$CACHE_INIT - INIT FILEREAD CACHE
               0180  ;++
               0180  ; FUNCTIONAL DESCRIPTION:
               0180  ;
               0180  ;       CACHE_INIT PERFORMS THE INITIALIZATION FOR THE FILEREAD CACHE
               0180  ;
               0180  ; CALLING SEQUENCE:
               0180  ;
               0180  ;       CALLG   ARGLIST,FIL$CACHE_INIT
               0180  ;
               0180  ; INPUT PARAMETERS:
               0180  ;
               0180  ;       CHANADR(AP)             ADDRESS TO RETURN LONG WORD CHANNEL
               0180  ;       FILNAM(AP)              ADDRESS OF DEVICE NAME STRING DESCRIPTOR
               0180  ;                               THE DEVICE NAME MUST CONTAIN THE ":"
               0180  ;                               IF THE ADDRESS IS 0, THE STRING IS NULL,
               0180  ;                               OR THE NAME DOES NOT CONTAIN A ":", THE
               0180  ;                               DEFAULT DEVICE NAME IS USED
               0180  ;       CACHE_SIZE(AP)          SIZE IN BYTES OF FILEREAD CACHE
               0180  ;       CACHE_ADR(AP)           ADDRESS OF FILEREAD CACHE
               0180  ;       DIR_CACHE_CNT(AP)       NUMBER OF DIRECTORY CACHE ENTRIES
               0180  ;       LBN_CACHE_CNT(AP)       NUMBER OF LBN CACHE ENTRIES
               0180  ;
               0180  ; IMPLICIT INPUTS:
               0180  ;
               0180  ;       NONE
               0180  ;
               0180  ; OUTPUT PARAMETERS:
               0180  ;
               0180  ;       R0 = ALWAYS SUCCESSFUL STATUS CODE
               0180  ;
               0180  ; IMPLICIT OUTPUTS:
               0180  ;
               0180  ;       FIL$GQ_CACHE QUAD WORD FILLED IN WITH SIZE AND ADDRESS OF CACHE
               0180  ;
               0180  ; COMPLETION CODES:
               0180  ;
               0180  ;       SS$_NORMAL              SUCCESSFUL COMPLETION
               0180  ;
               0180  ; SIDE EFFECTS:
               0180  ;
               0180  ;       NONE
               0180  ;
               0180  ;
               0180  ; EQUATED SYMBOLS, OFFSETS FROM AP
               0180  ;
     00000004  0180          CHANADR         =       4
     00000008  0180          FILNAM          =       8
     0000000C  0180          CACHE_SIZE      =       12
     00000010  0180          CACHE_ADR       =       16
     00000014  0180          DIR_CACHE_CNT   =       20
     00000018  0180          LBN_CACHE_CNT   =       24
               0180  ;
               0180  ;--
               0180  FIL$CACHE_INIT::
         0C3C  0180          .WORD   ^M<R2,R3,R4,R5,R10,R11>
               0182
```

```
                                  0182   566          ASSUME  CACHE_SIZE+4 EQ CACHE_ADR
               5A   0C AC    7D   0182   567          MOVQ    CACHE_SIZE(AP),R10          ;R10=SIZE, R11=ADR
     50   5A   00000218 8F   C3   0186   568          SUBL3   #FIL$C_SIZE,R10,R0          ;BYTES LEFT FOR DIR AND LBN CACHES
                        60   19   018E   569          BLSS    100$                        ;BRANCH IF NOT ENOUGH CACHE SPACE
                   6B   01   B0   0190   570          MOVW    #FIL$C_CACHE_ID,FIL$W_CACHE_ID(R11) ;SET CACHE ID
                                  0193   571                                              ;ALLOWS MOVING CACHES BETWEEN FILEREAD'S
       04 AB   00000218 8F   D0   0193   572          MOVL    #FIL$C_SIZE,FIL$L_DIROFF(R11) ;BEGINNING OF DIR CACHE
       08 AB   00000218 8F   D0   019B   573          MOVL    #FIL$C_SIZE,FIL$L_DIRNXT(R11) ;NEXT AVAILABLE SLOT IN DIR CACHE
          51      14 AC 24   C5   01A3   574          MULL3   #FIL$C_DIR_SIZE,DIR_CACHE_CNT(AP),R1 ;BYTE COUNT FOR DIR CACHE
                   50   51   C2   01A8   575          SUBL    R1,R0                       ;BYTE COUNT LEFT FOR LBN CACHE
                        43   19   01AB   576          BLSS    100$                        ;BRANCH IF NOT ENOUGH SPACE
  0C AB   51   00000218 8F   C1   01AD   577          ADDL3   #FIL$C_SIZE,R1,FIL$L_DIRMAX(R11) ;END OF DIR CACHE
                                  01B6   578
                                  01B6   579          ASSUME  FIL$L_DIRMAX EQ FIL$L_LBNOFF
            10 AB   0C AB    D0   01B6   580          MOVL    FIL$L_LBNOFF(R11),FIL$L_LBNNXT(R11) ;NEXT LBN ENTRY TO ALLOCATE
          51   18 AC    09   78   01BB   581          ASHL    #9,LBN_CACHE_CNT(AP),R1     ;BYTE COUNT IN LBN CACHE
                   50   51   D1   01C0   582          CMPL    R1,R0                       ;ENOUGH ROOM FOR WHOLE LBN CACHE
                        08   15   01C3   583          BLEQ    20$                         ;BRANCH IF YES
     51   50   000001FF 8F   CB   01C5   584          BICL3   #^X1FF,R0,R1                ;USE WHAT IS LEFT TRUNCATED
       14 AB   10 AB    51   C1   01CD   585 20$:     ADDL3   R1,FIL$L_LBNNXT(R11),FIL$L_LBNMAX(R11) ;END OF LBN CACHE
                      003D   30   01D3   586          BSBW    ASSIGN_DEV                  ;ASSIGN THE DEVICE
                     17 50   E9   01D6   587          BLBC    R0,100$                     ;BRANCH IF ERROR
                     18 AB   DF   01D9   588          PUSHAL  FIL$A_IXFHDR(R11)           ;ADDRESS TO READ INDEX FILE HEADER
                        7E   D4   01DC   589          CLRL    -(SP)                       ;UNUSED PARAMETER
                     04 BC   DD   01DE   590          PUSHL   @CHANADR(AP)                ;CHANNEL JUST ASSIGNED
             02D3'CF    03   FB   01E1   591          CALLS   #3,W^FIL$MOUNT              ;MOUNT THE VOLUME, RETURN INDEX FILE HDR
                     07 50   E9   01E6   592          BLBC    R0,100$                     ;BRANCH IF ERROR
         00000000'EF    5A   7D   01E9   593          MOVQ    R10,FIL$GQ_CACHE            ;SAVE DESCRIPTOR OF CACHE
                   50   01   D0   01F0   594 100$:    MOVL    S^#SS$_NORMAL,R0
                             04   01F3   595          RET
```

```
                            01F4        .SBTTL  FIL$CACHE_TRUNC - TRUNCATE FILEREAD CACHE
                            01F4  ;++
                            01F4  ; FUNCTIONAL DESCRIPTION:
                            01F4  ;
                            01F4  ;       CACHE_TRUNC TRUNCATES THE FILEREAD CACHE AND MAKES IT IMPOSSIBLE
                            01F4  ; TO ADD MORE DIRECTORY CACHE OR DIRECTORY LBN ENTRIES TO IT.  IN EFFECT
                            01F4  ; THIS ROUTINE TURNS THE CACHE INTO A READ-ONLY DATA BASE.
                            01F4  ;
                            01F4  ; CALLING SEQUENCE:
                            01F4  ;
                            01F4  ;       CALLG   ARGLIST,FIL$CACHE_TRUNC
                            01F4  ;
                            01F4  ; INPUT PARAMETERS:
                            01F4  ;
                            01F4  ;       NONE
                            01F4  ;
                            01F4  ; IMPLICIT INPUTS:
                            01F4  ;
                            01F4  ;       FIL$GQ_CACHE            DESCRIPTOR FOR THE CACHE
                            01F4  ;
                            01F4  ; OUTPUT PARAMETERS:
                            01F4  ;
                            01F4  ;       R0 = ALWAYS SUCCESSFUL STATUS CODE
                            01F4  ;
                            01F4  ; IMPLICIT OUTPUTS:
                            01F4  ;
                            01F4  ;       FIL$GQ_CACHE FILLED IN WITH ALTERED SIZE OF CACHE
                            01F4  ;
                            01F4  ; COMPLETION CODES:
                            01F4  ;
                            01F4  ;       SS$_NORMAL              SUCCESSFUL COMPLETION
                            01F4  ;
                            01F4  ; SIDE EFFECTS:
                            01F4  ;
                            01F4  ;       NONE
                            01F4  ;
                            01F4  ; EQUATED SYMBOLS
                            01F4  ;
                            01F4  ;
                            01F4  ;--
                            01F4
                            01F4  FIL$CACHE_TRUNC::
                      0000  01F4        .WORD   0
  50    00000004'EF    D0   01F6        MOVL    FIL$GQ_CACHE+4,R0       ;ADDRESS OF THE CACHE
     0C A0    08 A0    D0   01FD        MOVL    FIL$L_DIRNXT(R0),FIL$L_DIRMAX(R0) ;NO NEW DIRECTORY CACHE ENTRIES
     14 A0    10 A0    D0   0202        MOVL    FIL$L_LBNNXT(R0),FIL$L_LBNMAX(R0) ;NO MORE LBN BUFFERS
 00000000'EF  10 A0    D0   0207        MOVL    FIL$L_LBNNXT(R0),FIL$GQ_CACHE   ;SET NEW SIZE OF CACHE
          50    01     D0   020F        MOVL    S^#SS$_NORMAL,R0
                       04   0212        RET
```

```
                                0213         .SBTTL  ASSIGN_DEV - ASSIGN A DEVICE AND RETURN A CHANNEL
                                0213  ;++
                                0213  ; FUNCTIONAL DESCRIPTION:
                                0213  ;
                                0213  ;      ASSIGN THE GIVEN DEVICE AND RETURN A CHANNEL NUMBER
                                0213  ;      FOR BOOTING THIS VALUE IS A LONG WORD
                                0213  ;
                                0213  ; INPUTS:
                                0213  ;
                                0213  ;      FILNAM(AP)      = ADDRESS OF STRING DESCRIPTOR
                                0213  ;                        IF DEVICE NAME IS PRESENT IT MUST HAVE ":"
                                0213  ;      CHANADR(AP)     = ADDRESS TO RETURN LONG WORD OF CHANNEL NUMBER
                                0213  ;
                                0213  ; OUTPUTS:
                                0213  ;
                                0213  ;      R0,R1 ALTERED
                                0213  ;      R2 = SIZE OF DEVICE NAME STRING (DEFAULT IF NOT IN NAME STRING)
                                0213  ;           MAYBE 0 IF DEFAULT DEVICE STRING WAS NOT PRESENT
                                0213  ;           THIS IS THE CASE WHEN BOOTSTRAPPING.
                                0213  ;      R3 = ADDRESS OF DEVICE NAME STRING
                                0213  ;           0 IF DEFAULT DEVICE STRING WAS NOT PRESENT
                                0213  ;
                                0213  ;--
                                0213
                                0213  ASSIGN_DEV:
                    50     D4   0213         CLRL    R0                           ;ASSUME NULL DEFAULT DEVICE STRING
    51    00000000'EF      DE   0215         MOVAL   FIL$GT_DDDEV,R1              ;GET THE ADDRESS
                50  81     9A   021C         MOVZBL  (R1)+,R0                     ;SIZE OF DEFAULT DEVICE STRING
                    03     BB   021F  10$:   PUSHR   #^M<R0,R1>                   ;PUSH DEFAULT DEVICE DESCRIPTOR
                    52     7C   0221         CLRQ    R2                           ;ASSUME NULL STRING DESCRIPTOR
          50    08  AC     D0   0223         MOVL    FILNAM(AP),R0                ;ADDRESS OF STRING DESCRIPTOR
                    11     13   0227         BEQL    20$                          ;BRANCH IF NO NAME GIVEN
                52  60     7D   0229         MOVQ    (R0),R2                      ;R2 = SIZE, R3 = ADR OF FILE NAME STRING
       63   52  3A     3A   022C         LOCC    #^A/:/,R2,(R3)               ;ANY DEVICE SPECIFIED?
                    08     13   0230         BEQL    20$                          ;BRANCH IF NONE
             04 AE  53     D0   0232         MOVL    R3,4(SP)                     ;ADDRESS OF DEVICE NAME
          6E    51  53     C3   0236         SUBL3   R3,R1,(SP)                   ;SIZE OF DEVICE NAME STRING
                50  5E     D0   023A  20$:   MOVL    SP,R0                        ;ADDRESS OF DEV NAME DESCRIPTOR
                                023D         $ASSIGN_S (R0),@CHANADR(AP)          ;ASSIGM THE CHANNEL
                    0C     BA   024B         POPR    #^M<R2,R3>                   ;GET DEVICE NAME SIZE AND ADDRESS
                           05   024D         RSB
```

```
                        024E   689            .SBTTL  STORE3DIGITS - STORE 3 ASCII DIGITS
                        024E   690    ;++
                        024E   691    ; FUNCTIONAL DESCRIPTION:
                        024E   692    ;
                        024E   693    ;       STORE 3 DIGITS OF DIRECTORY STRING
                        024E   694    ;
                        024E   695    ; CALLING SEQUENCE:
                        024E   696    ;
                        024E   697    ;       BSBB    STORE3DIGITS
                        024E   698    ;
                        024E   699    ; INPUT:
                        024E   700    ;
                        024E   701    ;       R0 = NO. OF DIGITS TO PUT IN STRING
                        024E   702    ;       R1 = ADDRESS + 1 OF RIGHT MOST DIGIT
                        024E   703    ;       R2 = ADDRESS AT WHICH TO STORE 3 DIGITS
                        024E   704    ;
                        024E   705    ; OUTPUTS:
                        024E   706    ;
                        024E   707    ;       NONE
                        024E   708    ;
                        024E   709    ;--
                        024E   710
                        024E   711    STORE3DIGITS:
         03    50   D1  024E   712            CMPL    R0,#3                   ;3 DIGITS OR LESS SPECIFIED?
               03   15  0251   713            BLEQ    5$                      ;YES. BRANCH.
             0254   31  0253   714            BRW     BADFILNAM               ;NO. DIRECTORY STRING BAD. EXIT
                        0256   715                                            ;WITH ERROR.
   82    3030  8F   B0  0256   716    5$:     MOVW    #^A/00/,(R2)+           ;BACKGROUND WITH ASCII 0
         82    30   90  025B   717            MOVB    #^A/0/,(R2)+            ;
               03   11  025E   718            BRB     20$                     ;START LOOP AT BOTTOM
         72    71   90  0260   719    10$:    MOVB    -(R1),-(R2)             ;STORE BYTES LAST TO FIRST
                        0263   720                                            ;LEAVING LEADING ASCII 0'S
            FA 50   F4  0263   721    20$:    SOBGEQ  R0,10$                  ;LOOP ZERO OR MORE TIMES
                    05  0266   722            RSB
```

```
                             0267   724          .SBTTL  FORMDIRSTRING - GET A DIRECTORY STRING
                             0267   725 ;++
                             0267   726 ; FUNCTIONAL DESCRIPTION:
                             0267   727 ;
                             0267   728 ;        PULL THE FIRST DIRECTORY NAME OFF THE FRONT OF THE INPUT
                             0267   729 ;        DIRECTORY STRING AND FORM THE FULL FILE NAME OF THE DIRECTORY
                             0267   730 ;        TO LOOK UP.
                             0267   731 ;
                             0267   732 ; CALLING SEQUENCE:
                             0267   733 ;
                             0267   734 ;        BSBW    FORMDIRSTRING
                             0267   735 ;
                             0267   736 ; INPUTS:
                             0267   737 ;
                             0267   738 ;        R6 = SIZE OF DIRECTORY STRING
                             0267   739 ;        R7 = ADDRESS OF DIRECTORY STRING
                             0267   740 ;                THE STRING CONTAINS NO BRACKETS,
                             0267   741 ;                IT MAY BE OF THE FORM "DIR1.DIR2.DIR3...DIRN"
                             0267   742 ;                THE FIRST AND ONLY ITEM MAY BE IN THE FORM GROUP,MEMBER
                             0267   743 ;        DIRNAM(FP) = ADDRESS OF AREA TO BUILD THE NAME
                             0267   744 ;
                             0267   745 ; OUTPUTS:
                             0267   746 ;
                             0267   747 ;        R0 = SIZE OF DIRECTORY STRING
                             0267   748 ;        R1 = ADDRESS OF DIRECTORY STRING
                             0267   749 ;        R2,R3 PRESERVED
                             0267   750 ;        R6,R7 UPDATED TO POINT AT THE REST OF THE STRING
                             0267   751 ;--
                             0267   752
                             0267   753 FORMDIRSTRING:
      67    56    2E    3A   0267   754          LOCC    #^A/./,R6,(R7)          ;FIND NEXT DIRECTORY STRING
      56    50    01    C3   026B   755          SUBL3   #1,R0,R6                ;SIZE OF REST, SKIP THE "."
                             026F   756                                          ;-1 IF EMPTY
      50    51    57    C3   026F   757          SUBL3   R7,R1,R0                ;BYTE COUNT OF DIRECTORY NAME
            51    57    D0   0273   758          MOVL    R7,R1                   ;ADDRESS OF DIRECTORY NAME
      57    01  A140    9E   0276   759          MOVAB   1(R1)[R0],R7            ;ADDRESS OF NEXT BYTE BEYOND "."
                  07    BB   027B   760          PUSHR   #^M<R0,R1,R2>           ;SAVE STRING DESCRIPTORS AND R2
            09    50    D1   027D   761          CMPL    R0,#9                   ;LENGTH OF DIRECTORY STRING OKAY?
                  03    15   0280   762          BLEQ    5$                      ;YES. BRANCH.
                0225    31   0282   763          BRW     BADFILNAM               ;NO. EXIT WITH ERROR.
      61    50    2C    3A   0285   764 5$:      LOCC    #^A/,/,R0,(R1)          ;GOOD STRING: ANY ","?
                  39    13   0289   765          BEQL    20$                     ;BRANCH IF NOT, RETURN THE DESCRIPTOR AS IS
                  50    DD   028B   766          PUSHL   R0                      ;SAVE REMAINING BYTE COUNT
 50    04 AE      50    C3   028D   767          SUBL3   R0,4(SP),R0             ;BYTE COUNT TO LEFT OF ","
      52    EA    AD    DE   0292   768          MOVAL   DIRNAM(FP),R2           ;ADDRESS TO STORE FIRST 3 CHARS
                  B6    10   0296   769          BSBB    STORE3DIGITS            ;STORE THEM
      50    8E    01    C3   0298   770          SUBL3   #1,(SP)+,R0             ;COUNT OF CHARS TO RIGHT OF ","
      51    8E    8E    C1   029C   771          ADDL3   (SP)+,(SP)+,R1          ;ADR OF BYTE TO RIGHT OF LAST CHAR
      52    ED    AD    DE   02A0   772          MOVAL   DIRNAM+3(FP),R2         ;ADR TO STORE LAST 3 CHARS OF DIR NAME
                  A8    10   02A4   773          BSBB    STORE3DIGITS            ;STORE THEM
                  04    BA   02A6   774          POPR    #^M<R2>                 ;RESTORE SAVED R2
            50    06    D0   02A8   775          MOVL    #6,R0                   ;6 BYTES STRING SIZE
      51    EA  AD40    9E   02AB   776 10$:     MOVAB   DIRNAM(FP)[R0],R1       ;POINT TO END OF STRING
 81   5249442E    8F    D0   02B0   777          MOVL    #^A/.DIR/,(R1)+         ;PUT TYPE IN STRING
      61    313B  8F    B0   02B7   778          MOVW    #^A/;1/,(R1)            ;AND VERSION AS WELL
      51    EA    AD    9E   02BC   779          MOVAB   DIRNAM(FP),R1           ;ADDRESS OF STRING
            50    06    C0   02C0   780          ADDL    #6,R0                   ;SIZE INCLUDES ".DIR;1"
```

```
                           05  02C3   781          RSB
                     38    BB  02C4   782 20$:     PUSHR   #^M<R3,R4,R5>              ;SAVE THESE FROM MOVC3
                               02C6   783 ;
                               02C6   784 ; 12(SP) = SIZE OF STRING, 16(SP) = ADDRESS
                               02C6   785 ;
      EA AD   10 BE    0C AE   28  02C6   786          MOVC3   12(SP),@16(SP),DIRNAM(FP) ;MOVE NAME TO SCRATCH AREA
                           38  BA  02CD   787          POPR    #^M<R3,R4,R5>              ;RESTORE REGISTERS
                           07  BA  02CF   788          POPR    #^M<R0,R1,R2>
                           D8  11  02D1   789          BRB     10$
```

```
                         02D3          .SBTTL  MOUNT - MOUNT THE VOLUME, INIT FOR FILE LOOKUP
                         02D3   ;++
                         02D3   ; FUNCTIONAL DESCRIPTION:
                         02D3   ;
                         02D3   ;       MOUNT PERFORMS THE NECESSARY INITIALIZATION FOR FILE LOOKUP.
                         02D3   ; IT READS THE HOME BLOCK, AND THEN RETURNS THE INDEX FILE HEADER TO THE
                         02D3   ; SPECIFIED BUFFER.  THE INDEX FILE HEADER IS ALTERED BY RECORDING THE
                         02D3   ; VIRTUAL BLOCK OFFSET REQUIRED TO TRANSLATE 'FILE NUMBER' TO INDEX FILE VBN
                         02D3   ;
                         02D3   ; CALLING SEQUENCE:
                         02D3   ;
                         02D3   ;       CALLG   ARGLIST,FIL$MOUNT
                         02D3   ;
                         02D3   ; INPUT PARAMETERS:
                         02D3   ;
                         02D3   ;       CHAN(AP)                        CHANNEL ON WHICH DEVICE IS ASSIGNED
                         02D3   ;       UNUSED                          2ND PARAMETER NOT USED
                         02D3   ;       IXFHDR(AP)                      ADDRESS TO RETURN INDEX FILE HEADER
                         02D3   ;
                         02D3   ; IMPLICIT INPUTS:
                         02D3   ;
                         02D3   ;       NONE
                         02D3   ;
                         02D3   ; OUTPUT PARAMETERS:
                         02D3   ;
                         02D3   ;       R0 = SYSTEM STATUS CODE
                         02D3   ;
                         02D3   ; IMPLICIT OUTPUTS:
                         02D3   ;
                         02D3   ;       NONE
                         02D3   ;
                         02D3   ; COMPLETION CODES:
                         02D3   ;
                         02D3   ;       SS$_NORMAL                      SUCCESSFUL COMPLETION
                         02D3   ;       SS$_FILESTRUCT                  FILE STRUCTURE LEVEL NOT SUPPORTED
                         02D3   ;       SS$_BADCHKSUM                   CHECKSUM ERROR ON HOME BLOCK OR INDEX FILE HEADER
                         02D3   ;       SS$_BADFILEHDR                  INDEX FILE HEADER IS BAD
                         02D3   ;
                         02D3   ; SIDE EFFECTS:
                         02D3   ;
                         02D3   ;       NONE
                         02D3   ;
                         02D3   ;
                         02D3   ; EQUATED SYMBOLS, OFFSETS FROM AP
                         02D3   ;
           00000004      02D3          CHAN            =       4
           0000000C      02D3          IXFHDR          =       12
                         02D3   ;
                         02D3   ;--
                         02D3
                         02D3   FIL$MOUNT::
               001C      02D3          .WORD   ^M<R2,R3,R4>
      53   0C AC   D0    02D5          MOVL    IXFHDR(AP),R3                   ;ADDRESS OF BUFFER
   7E  01     09   9C    02D9          ROTL    #9,#1,-(SP)                     ;NUMBER OF BYTES TO READ
       7E     21   3C    02DD          MOVZWL  #IO$_READLBLK,-(SP)             ;READ LOGICAL BLOCK FUNCTION
              53   DD    02E0          PUSHL   R3                              ;BUFFER ADDRESS
              01   DD    02E2          PUSHL   #1                              ;LOGICAL BLOCK NUMBER 1 IS HOME BLK
              04 AC     DD  02E4          PUSHL   CHAN(AP)                  ;CHANNEL
                 05     DD  02E7          PUSHL   #5                        ;NO. OF ARGUMENTS
        0000'CF  6E     FA  02E9          CALLG   (SP),W^FIL$RDWRTLBN       ;READ THE HOME BLOCK
              60 50     E9  02EE          BLBC    R0,30$                    ;BRANCH IF ERROR
           51    53     D0  02F1          MOVL    R3,R1                     ;ADDRESS OF HOME BLOCK
           50    1D     3C  02F4          MOVZWL  #HM2$W_CHECKSUM1@-1,R0    ;NO. OF WORDS IN FIRST CHECKSUM
               04B7     30  02F7          BSBW    FIL$CHECKSUM1             ;CHECK THE FIRST CHECKSUM
           51    53     D0  02FA          MOVL    R3,R1                     ;ADR OF HOME BLOCK AGAIN
               04AC     30  02FD          BSBW    FIL$CHECKSUM              ;CHECK THE MAIN CHECKSUM
        02    0D A3     91  0300          CMPB    HM2$B_STRUCLEV(R3),#2     ;IS THIS STRUCTURE LEVEL 2?
                 4C     12  0304          BNEQ    40$                       ;BR IF NOT STRUCTURE LEVEL 2
                            0306     ;
                            0306     ; STRUCTURE LEVEL 2
                            0306     ;
           51    18 A3  D0  0306          MOVL    HM2$L_IBMAPLBN(R3),R1     ;INDEX BIT MAP STARTING LBN
           50    20 A3  3C  030A          MOVZWL  HM2$W_IBMAPSIZE(R3),R0    ;INDEX BIT MAP SIZE IN BLOCKS
           54    1C A3  D0  030E          MOVL    HM2$L_MAXFILES(R3),R4     ;MAXIMUM FILES ON VOLUME
                            0312                                            ;ONLY INTERESETED IN HIGH 16 BITS
     54    0E A3    04  A5  0312          MULW3   #4,HM2$W_CLUSTER(R3),R4   ;4*CLUSTER TO LOW WORD OF R4
     54    0E A3    04  A5  0317          MULW3   #4,HM2$W_CLUSTER(R3),R4   ;4*CLUSTER TO LOW WORD OF R4
              54    50  A0  031C          ADDW    R0,R4                     ;LOW WORD IS VBNOFFSET
                            031F                                            ;FROM FILE ID TO INDEX FILE VBN
                            031F     ;
                            031F     ; READ INDEX FILE HEADER
                            031F     ;        R0 - NUMBER OF BLOCKS IN INDEX FILE
                            031F     ;        R1 - STARTING LBN OF INDEX FILE
                            031F     ;
     08 AE    51    50  C1  031F          ADDL3   R0,R1,8(SP)               ;DESIRED LBN TO ARG LIST
        0000'CF     8E  FB  0324          CALLS   (SP)+,W^FIL$RDWRTLBN      ;READ INDEX FILE HEADER
                            0329                                            ;STRIP OFF THE ARGUMENT LIST
              25 50     E9  0329          BLBC    R0,30$                    ;BRANCH IF ERROR
           51    53     D0  032C          MOVL    R3,R1                     ;ADDRESS OF HEADER
                 7E     D4  032F          CLRL    -(SP)                     ;FORM FILE ID ON STACK
       00010001 8F      DD  0331          PUSHL   #^X10001                  ;FOR THE INDEX FILE HEADER
           50    5E     D0  0337          MOVL    SP,R0                     ;ADDRESS OF FILE ID
               043F     30  033A          BSBW    FIL$CHKFILHDR             ;CHECK THE FILE HEADER (SEE IF
                            033D                                            ;FILE IDS MATCH)
     01FE C3    54      B0  033D          MOVW    R4,FH2$W_VBNOFFSET(R3)    ;STORE VBN OFFSET
                            0342     ;
                            0342     ; IF MAXFILES WAS GREATER THAN ^XFFFF THEN HIGH 16 BITS OF R4 WILL BE
                            0342     ; NON-ZERO.  IN THIS CASE, RECORD THE BIGFILNUM BIT IN THE STRUCLEV WORD
                            0342     ;
     54    54     F0 8F  78  0342         ASHL    #-16,R4,R4                ;SEE IF HIGH 16 BITS = 0
                     05  13  0347         BEQL    25$
      00 06 A3       0A  E2  0349         BBSS    #FH2$V_BIGFILNUM,FH2$W_STRUCLEV(R3),25$ ;MUST USE HIGH 8 BITS
                             034E                                           ;OF RVN FIELD AS FILE NUMBER EXTENSION
           50    01     3C  034E     25$: MOVZWL  #SS$_NORMAL,R0            ;SUCCESSFUL COMPLETION
                        04  0351     30$: RET
     50    08C0 8F      3C  0352     40$: MOVZWL  #SS$_FILESTRUCT,R0        ;UNSUPPORTED FILE STRUCTURE LEVEL
                        04  0357          RET
```

```
0358            .SBTTL  FINDFILID - FIND FILE ID FOR SPECIFIED FILE
0358    ;++
0358    ; FUNCTIONAL DESCRIPTION:
0358    ;
0358    ;       FINDFILID SCANS A SPECIFIED DIRECTORY FOR A FILE AND
0358    ; RETURNS ITS FILE ID IF FOUND.  STRUCTURE LEVEL 2 DIRECTORIES
0358    ; ARE SUPPORTED, 0 VERSION NUMBER MEANS FIND MOST RECENT VERSION,
0358    ; -1 VERSION (FIND OLDEST) IS NOT SUPPORTED.
0358    ;
0358    ; NOTE THAT NON-CONTIGUOUS DIRECTORIES ARE NOT SUPPORTED.
0358    ;
0358    ; CALLING SEQUENCE:
0358    ;
0358    ;       CALLG   ARGLIST,FIL$FINDFILID
0358    ;
0358    ; INPUT PARAMETERS:
0358    ;
0358    ;       CHAN(AP)                        ;CHANNEL ON WHICH DEVICE IS ASSIGNED
0358    ;       FILDSC(AP)      =               ADDRESS OF 3 LONG WORD FILE DESCRIPTOR
0358    ;                                       1 - SIZE OF ASCII STRING,  A 0 VALUE MEANS
0358    ;                                               USE THE CONTENTS OF THE NAMBLK BELOW
0358    ;                                       2 - ADDRESS OF ASCII STRING
0358    ;                               3 - ADDRESS OF NAME BLOCK - (OBSOLETE, LEVEL 1 ONLY)
0358    ;                                               MAY CONTAIN DEFAULTS, BUT MUST BE
0358    ;                                               AT LEAST INITIALIZED TO ZERO
0358    ;                                               IT WILL BE WRITTEN.
0358    ;       IXFHDR(AP)                      ADR OF INDEX FILE HDR AS RETURNED FROM FIL$MOUNT
0358    ;       DIRBUF(AP)                      ADR OF 512 BYTE BUFFER TO USE FOR DIRECTORY SCAN
0358    ;       STATBLK(AP)                     ADDRESS OF 2 LONG WORD AREA USED FOR A
0358    ;                                       SCRATCH STATISTICS BLOCK
0358    ;       FILID(AP)                       ADR OF 3 WORD AREA USED BOTH AS THE ID OF
0358    ;                                       THE DIRECTORY TO SCAN AND AS THE PLACE TO
0358    ;                                       RETURN THE ID OF THE FILE FOUND
0358    ;
0358    ; IMPLICIT INPUTS:
0358    ;
0358    ;       NONE
0358    ;
0358    ; OUTPUT PARAMETERS:
0358    ;
0358    ;       R0 = SYSTEM STATUS CODE
0358    ;
0358    ; IMPLICIT OUTPUTS:
0358    ;
0358    ;       NONE
0358    ;
0358    ; COMPLETION CODES:
0358    ;
0358    ;       SS$_NORMAL                      SUCCESSFUL COMPLETION
0358    ;       SS$_NOSUCHFILE                  FILE NOT FOUND
0358    ;       SS$_BADFILENAME                 SYNTAX ERROR IN FILE NAME
0358    ;       SS$_BADCHKSUM                   CHECKSUM ERROR ON DIRECTORY FILE HEADER
0358    ;       SS$_BADFILEHDR                  DIRECTORY FILE HEADER WAS BAD
0358    ;
0358    ; SIDE EFFECTS:
0358    ;
0358    ;       NONE
```

```
                                      0358   956 ;
                                      0358   957 ;
                                      0358   958 ; EQUATED SYMBOLS, OFFSETS FROM AP
                                      0358   959 ;
                            00000004  0358   960              CHAN                     =            4
                            00000008  0358   961              FILDSC                   =            8
                            0000000C  0358   962              IXFHDR                   =           12
                            00000010  0358   963              DIRBUF                   =           16
                            00000014  0358   964              STATBLK                  =           20
                            00000018  0358   965              FILID                    =           24
                                      0358   966 ;
                                      0358   967 ; OFFSETS FROM FP
                                      0358   968 ;
                                      0358   969              $OFFSET 0,NEGATIVE,<-
                                      0358   970              DIR_BFCNT,-                                     ;BUFFER COUNT REMAINING IN LBN CACHE
                                      0358   971              DIR_BUF,-                                       ;NEXT BUFFER ADDRESS IN DIR LBN CACHE
                                      0358   972              ENTRY_ADR,-                                     ;FOUND CACHE ENTRY ADR
                                      0358   973              <ENTRY,FIL$C_DIR_SIZE>,-                        ;CACHE ENTRY FOR SEARCH/CREATE
                                      0358   974              <SCRATCH_SIZE,0>-                               ;SIZE OF SCRATCH AREA
                                      0358   975              >
                                      FFFC       DIR_BFCNT:
                                      FFF8       DIR_BUF:
                                      FFF4       ENTRY_ADR:
                                      FFD0       ENTRY:
                                      FFD0       SCRATCH_SIZE:
                                      0358   976 ;
                                      0358   977 ;--
                                      0358   978
                                      0358   979 FIL$FINDFILID::
                                0FFC  0358   980              .WORD      ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                       5E   30    C2  035A   981              SUBL       #-SCRATCH_SIZE,SP                   ;ALLOCATE SCRATCH SPACE
    6E    30    00    6E    00    2C  035D   982              MOVC5      #0,(SP),#0,#-SCRATCH_SIZE,(SP) ;ZERO THE SCRATCH STORAGE
                            5B    D4  0363   983              CLRL       R11                                 ;ASSUME NO CACHE
                      02    6C    D1  0365   984              CMPL       (AP),#2                             ;IF ONLY 2 ARGUMENTS
                            08    12  0368   985              BNEQ       5$
                    5B   04 AC    D0  036A   986              MOVL       4(AP),R11                           ;THE FIRST IS THE CACHE ADDRESS
                    5C   08 AC    D0  036E   987              MOVL       8(AP),AP                            ;THE SECOND IS THE REAL ARGUMENT LIST
                            5B    D5  0372   988 5$:          TSTL       R11                                 ;CACHE ENABLED?
                            66    13  0374   989              BEQL       65$                                 ;BRANCH IF NOT
                                      0376   990 ;
                                      0376   991 ; WE DO HAVE A CACHE TO LOOK IN AND MAKE ENTRIES IN, SET UP THE
                                      0376   992 ; SCRATCH REGION FOR A LOOKUP
                                      0376   993 ;
                                      0376   994              ASSUME     ENTRY EQ SCRATCH_SIZE               ;SCRATCH CACHE ENTRY MUST BE ON TOP
                                      0376   995              ASSUME     FIL$A_DIR_FID EQ 0                  ;DIR ID IS AT FRONT OF CACHE ENTRY
             6E    18 BC    06    28  0376   996              MOVC3      #6,@FILID(AP),(SP)                  ;STORE DIRECTORY ID
                   50    08 BC    7D  037B   997              MOVQ       @FILDSC(AP),R0                      ;GET LOOKUP NAME DESCRIPTOR
                      06    50    D1  037F   998              CMPL       R0,#6                               ;MUST BE MORE THAN ".DIR;1"
                            25    15  0382   999              BLEQ       10$                                 ;BRANCH IF NOT A DIRECTORY NAME
                      0F    50    D1  0384  1000              CMPL       R0,#15                              ;AT MOST 9 CHAR WITH ".DIR;1"
                            20    14  0387  1001              BGTR       10$                                 ;BRANCH IF NOT A DIRECTORY NAME
                52    51    50    C1  0389  1002              ADDL3      R0,R1,R2                            ;POINT OFF END OF NAME STRING
                  72   313B 8F    B1  038D  1003              CMPW       #^A/;1/,-(R2)                       ;LAST 2 BYTES ";1" ?
                            15    12  0392  1004              BNEQ       10$                                 ;BRANCH IF NOT A DIRECTORY NAME
              72   5249442E 8F    D1  0394  1005              CMPL       #^A/.DIR/,-(R2)                     ;PRECEDED BY ".DIR" ?
                            0C    12  039B  1006              BNEQ       10$                                 ;BRANCH IF NOT A DIRECTORY NAME
                      50    06    C2  039D  1007              SUBL       #6,R0                               ;JUST KEEP THE NAME PART
```

```
          D6 AD   50   90  03A0  1008          MOVB    R0,FIL$T_DIR_NAM+ENTRY(FP) ;PUT SIZE AND NAME STRING
    D7 AD    61   50   28  03A4  1009          MOVC3   R0,(R1),FIL$T_DIR_NAM+1+ENTRY(FP) ;IN ENTRY TO LOOKUP
    58    5B   04 AB   C1  03A9  1010 10$:     ADDL3   FIL$L_DIROFF(R11),R11,R8 ;ADDRESS OF DIRECTORY CACHE
    59    5B   08 AB   C1  03AE  1011          ADDL3   FIL$L_DIRNXT(R11),R11,R9 ;ADDRESS OF LAST+1 BYTE
                  1E   11  03B3  1012          BRB     60$                      ;LOOP 0 OR MORE TIMES
                           03B5  1013
                           03B5  1014          ASSUME  FIL$T_DIR_NAM EQ FIL$A_DIR_FID+6
      68   D0 AD   10   29  03B5  1015 20$:     CMPC3   #6+10,FIL$A_DIR_FID+ENTRY(FP), - ;DOES FID AND NAME MATCH?
                           03BA  1016                  FIL$A_DIR_FID(R8)
                  06   12  03BA  1017          BNEQ    30$                      ;BRANCH IF DIDN'T MATCH ALL OF IT
          F4 AD   58   D0  03BC  1018          MOVL    R8,ENTRY_ADR(FP)         ;RECORD THAT A MATCH WAS FOUND
                  1E   11  03C0  1019          BRB     70$
                           03C2  1020 ;
                           03C2  1021 ; FAILED TO MATCH THE ENTIRE ENTRY, DID WE MATCH THE DIR ID FIELD?
                           03C2  1022 ;
             0A   50   D1  03C2  1023 30$:     CMPL    R0,#10                   ;IF 10 OR LESS CHAR'S LEFT
                  09   14  03C5  1024          BGTR    50$                      ;THEN MATCHED THE DIR ID
          F4 AD   58   D0  03C7  1025          MOVL    R8,ENTRY_ADR(FP)         ;SAVE THIS PARTIAL MATCH
             D6 AD   95    03CB  1026          TSTB    FIL$T_DIR_NAM+ENTRY(FP)  ;IF NOT SEARCHING FOR A DIR NAME
                  10   13  03CE  1027          BEQL    70$                      ;THEN THIS ENTRY WILL DO FINE
             58   24   C0  03D0  1028 50$:     ADDL    #FIL$C_DIR_SIZE,R8       ;ADDRESS OF NEXT CACHE ENTRY
             59   58   D1  03D3  1029 60$:     CMPL    R8,R9                    ;DONE SCANNING DIR CACHE?
                  DD   1F  03D6  1030          BLSSU   20$                      ;BRANCH IF NOT, CHECK NEXT ENTRY
                           03D8  1031 ;
                           03D8  1032 ; ENTRY_ADR(FP) = ADDRESS OF CACHE HIT ENTRY
                           03D8  1033 ;               = 0 IF NO MATCH FOUND
                           03D8  1034 ; IF WE DROP THROUGH TO HERE AND WE GOT A CACHE HIT, THEN IT WAS
                           03D8  1035 ; NOT EXACTLY WHAT WE WERE LOOKING FOR.  BUT IT DID MATCH THE DIRECTORY.
                           03D8  1036 ;
          58   F4 AD   D0  03D8  1037          MOVL    ENTRY_ADR(FP),R8         ;WAS THERE A CACHE HIT?
                  3E   13  03DC  1038 65$:     BEQL    READ_DIR_HEADER          ;BRANCH IF NO
                  0F   11  03DE  1039          BRB     80$                      ;YES, FOR DIRECTORY LBN AND SIZE
                           03E0  1040 ;
                           03E0  1041 ; FOUND WHAT WE WERE LOOKING FOR - MAY ONLY NEED DIRECTORY LBN AND SIZE
                           03E0  1042 ;
             1E A8     D5  03E0  1043 70$:     TSTL    FIL$A_DIR_OFID(R8)       ;DID WE GET A FILE ID?
                  0A   13  03E3  1044          BEQL    80$                      ;BRANCH IF NOT
   18 BC     1E A8  06  28  03E5  1045          MOVC3   #6,FIL$A_DIR_OFID(R8),@FILID(AP) ;RETURN THE FILE ID
             50   01   3C  03EB  1046          MOVZWL  S^#SS$_NORMAL,R0         ;SET SUCCESS STATUS
                       04  03EE  1047 72$:     RET                              ;AND RETURN
                           03EF  1048 ;
                           03EF  1049 ; CACHE HIT ONLY FOUND THE DIRECTORY LBN AND SIZE, SAVING THE
                           03EF  1050 ; READ OF THE DIRECTORY FILE HEADER.
                           03EF  1051 ;
      E0 AD     10 A8   7D  03EF  1052 80$:     MOVQ    FIL$Q_DIR_HDR(R8),FIL$Q_DIR_HDR+ENTRY(FP) ;SAVE DIRHDR
                           03F4  1053                                           ;INFO FOR MAKING A NEW ENTRY
                           03F4  1054                                           ;WITH THE DIRECTORY FID IN IT
      E8 AD     18 A8   D0  03F4  1055          MOVL    FIL$L_DIR_BFOFF(R8),FIL$L_DIR_BFOFF+ENTRY(FP)
      EC AD     1C A8   B0  03F9  1056          MOVW    FIL$W_DIR_BFCNT(R8),FIL$W_DIR_BFCNT+ENTRY(FP)
                           03FE  1057                                           ;SAVE DIR LBN CACHE INFO TOO
                           03FE  1058 ;
                           03FE  1059 ; AT THIS POINT ENTRY(FP) CONTAINS DIRECTORY LBN CACHE INFORMATION
                           03FE  1060 ; IF ONE HAD ALREADY EXISTED OR IF WE JUST CREATED IT.
                           03FE  1061 ; SET UP THE WORKING LOCATIONS FOR THE DIRECTORY LBN CACHE
                           03FE  1062 ;
    56    E4 AD    01   C3  03FE  1063 90$:     SUBL3   #1,FIL$L_DIR_LBN+ENTRY(FP),R6 ;R6=STARTING LBN - 1
       57    E0 AD  3C      0403  1064          MOVZWL  FIL$W_DIR_BKCNT+ENTRY(FP),R7 ;R7=SIZE OF DIRECTORY IN BLOCKS
```

```
        FC AD    EC AD   3C  0407          MOVZWL  FIL$W_DIR_BFCNT+ENTRY(FP),DIR_BFCNT(FP) ;BUFFER COUNT IN LBN CACHE
 F8 AD   5B      E8 AD   C1  040C          ADDL3   FIL$L_DIR_BFOFF+ENTRY(FP),R11,DIR_BUF(FP) ;STARTING ADR IN CACHE
            57      56   C0  0412          ADDL    R6,R7                         ;LAST LBN OF FILE INCLUSIVE
                  00A4   31  0415          BRW     FIND_LEVEL2_1
                             0418  BADDIR2:
                  0089   31  0418          BRW     BADDIR
                             041B  BADRET1:
                    04       041B          RET
                             041C  ;
                             041C  ; CACHE WAS NOT ENABLED OR THERE WAS NOT A HIT FOR THIS DIRECTORY
                             041C  ;
                             041C  READ_DIR_HEADER:
     05BA'CF    6C       FA  041C          CALLG   (AP),W^FIL$RDCHKFILHDR        ;READ AND CHECK DIRECTORY FILE HEADER
             F7 50       E9  0421          BLBC    R0,BADRET1                    ;BRANCH IF ERROR
        55    10 AC      D0  0424          MOVL    DIRBUF(AP),R5                 ;ADDRESS OF BUFFER TO READ DIRECTORY BLOCKS
   56      14 BC    01   C3  0428          SUBL3   #1,@STATBLK(AP),R6            ;GET START LBN - 1
                             042D  ;
                             042D  ; IF THIS RESULT IS NEGATIVE, THEN THE DIRECTORY WAS NOT CONTIGUOUS.
                             042D  ; THIS CODE SUPPORTS ONLY CONTIGUOUS DIRECTORIES, ANOTHER BUFFER WOULD
                             042D  ; BE REQUIRED TO HOLD THE DIRECTORY HEADER IN ORDER TO SUPPORT NON-CONTIGUOS
                             042D  ; DIRECTORIES.  SUCH DIRECTORIES ARE ONLY CREATED BY FILES-11 WHEN
                             042D  ; A DIRECTORY MUST BE EXTENDED AND THERE IS NOT ENOUGH CONTIGUOUS SPACE
                             042D  ; ANYWHERE ON THE VOLUME TO MAKE A NEW DIRECTORY OF THE CORRECT SIZE.
                             042D  ;
                    E9   19  042D          BLSS    BADDIR2                       ;BRANCH IF NOT CONTIGUOUS
                             042F  ;
                             042F  ; SEE IF THIS LOOKS LIKE A DIRECTORY FILE
                             042F  ;
        54      14 A5    DE  042F          MOVAL   FH2$W_RECATTR(R5),R4          ;ADDRESS OF LEVEL 2 RECORD ATTRIBUTES
   57      08 A4    10   9C  0433  10$:    ROTL    #16,FAT$L_EFBLK(R4),R7        ;VBN OF DIRECTORY EOF
                 0C A4   B5  0438          TSTW    FAT$W_FFBYTE(R4)              ;IF ZERO, EFBLK IS LAST+1 VBN
                    02   12  043B          BNEQ    20$
                    57   D7  043D          DECL    R7                            ;CORRECT TO GET LAST VBN
   E4 AD      56    01   C1  043F  20$:    ADDL3   #1,R6,FIL$L_DIR_LBN+ENTRY(FP) ;SAVE START LBN,
        E0 AD       57   B0  0444          MOVW    R7,FIL$W_DIR_BKCNT+ENTRY(FP)  ;DIRECTORY SIZE,
        E2 AD       01   90  0448          MOVB    #1,FIL$B_DIR_LVL+ENTRY(FP)    ;AND INDIC ODS-2 STRUCTURE LEVEL
                             044C                                                ;(THIS IS FOR BACKWARD-COMPAT W/SYSBOOT)
                             044C  ;
                             044C  ; SEE IF WE CAN SET UP A CACHE OF THE DIRECTORY BLOCKS FOR THIS DIRECTORY
                             044C  ;
                    5B   D5  044C          TSTL    R11                           ;ANY CACHEING ENABLED?
                    4F   13  044E          BEQL    80$                           ;BRANCH IF NOT
  52    14 AB    10 AB   C3  0450          SUBL3   FIL$L_LBNNXT(R11),FIL$L_LBNMAX(R11),R2 ;NO. OF BYTES
                             0456                                                ;AVAILABLE TO ALLOCATE
   52     52     F7 8F   78  0456          ASHL    #-9,R2,R2                     ;NO. OF PAGES AVAILABLE
                    42   13  045B          BEQL    80$                           ;BRANCH IF NO SPACE AT ALL
            57      52   D1  045D          CMPL    R2,R7                         ;ENOUGH ROOM FOR WHOLE DIR
                    03   15  0460          BLEQ    40$                           ;BRANCH IF NOT, USE WHAT IS LEFT
            52      57   D0  0462          MOVL    R7,R2                         ;YES, USE THE RIGHT SIZE
        53       10 AB   D0  0465  40$:    MOVL    FIL$L_LBNNXT(R11),R3          ;OFFSET TO DIR LBN CACHE
                             0469  ;
                             0469  ; READ THE DISK BLOCKS INTO THE LBN CACHE.
                             0469  ;
        7E    52    09   78  0469          ASHL    #9,R2,-(SP)                   ;BYTE COUNT TO TRANSFER
              7E    21   3C  046D          MOVZWL  #IO$_READLBLK,-(SP)           ;READ LOGICAL BLOCK FUNCTION
                  6B43   9F  0470          PUSHAB  (R11)[R3]                     ;BUFFER ADDRESS
                 E4 AD   DD  0473          PUSHL   FIL$L_DIR_LBN+ENTRY(FP)       ;STARTING LBN
```

```
          04 AC    DD  0476            PUSHL   CHAN(AP)                 ;CHANNEL
   0000'CF    05   FB  0479            CALLS   #5,W^FIL$RDWRTLBN        ;FILL THE DIR LBN CACHE
          1E 50    E9  047E            BLBC    R0,80$                   ;BRANCH IF ERROR
                       0481      ;
                       0481      ; NOTE THAT DIRECTORY BLOCKS ARE IN MEMORY
                       0481      ;
    FC AD     52   D0  0481            MOVL    R2,DIR_BFCNT(FP)         ;COUNT OF BLOCKS READ IN
   F8 AD    6B43   9E  0485            MOVAB   (R11)[R3],DIR_BUF(FP)    ;ADDRESS OF FIRST BLOCK READ IN
          D6 AD    95  048A            TSTB    FIL$T_DIR_NAM+ENTRY(FP)  ;ARE WE LOOKING UP ANOTHER DIRECTORY?
             10    12  048D            BNEQ    80$                      ;BRANCH IF YES, DON'T ALLOCATE
                       048F                                             ;A PERMANENT CACHE ENTRY FOR
                       048F                                             ;AN INTERMEDIATE LEVEL DIRECTORY
    EC AD     52   B0  048F            MOVW    R2,FIL$W_DIR_BFCNT+ENTRY(FP) ;SET UP FOR PERMANENT ENTRY
    E8 AD     53   D0  0493            MOVL    R3,FIL$L_DIR_BFOFF+ENTRY(FP) ;SAVE THE SIZE AND OFFSET OF CACHE
   51   52    09   78  0497            ASHL    #9,R2,R1                 ;NO. OF BYTES IN CACHE
     10 AB    51   C0  049B            ADDL    R1,FIL$L_LBNNXT(R11)     ;ALLOCATE THE CACHE
                       049F      ;
                       049F      ; IF IT WAS POSSIBLE TO READ THE DIRECTORY INTO THE LBN CACHE AREA,
                       049F      ; THE SCAN WILL FIND THESE BLOCKS AS THEY ARE NEEDED.
                       049F      ;
                       049F      80$:
                       049F      ;
                       049F      ; R6 = STARTING LBN - 1 FOR THE DIRECTORY
                       049F      ; R7 = COUNT OF BLOCKS OF DIRECTORY TO BE SCANNED
                       049F      ;
          57  56   C0  049F            ADDL    R6,R7                            ;LAST LBN OF FILE (INCLUSIVE)
              0C   11  04A2            BRB FIND_LEVEL2
                       04A4
                       04A4      ;
                       04A4      ; ERROR RETURNS.
                       04A4      ;
   50   0828 8F    3C  04A4      BADDIR: MOVZWL  #SS$_BADIRECTORY,R0
                   04  04A9      BADRET: RET
                       04AA      BADFILNAM:
   50   0818 8F    3C  04AA            MOVZWL  #SS$_BADFILENAME,R0      ;RETURN ERROR CODE.
                   04  04AF            RET
```

```
                           04B0  1159          .SBTTL  FIL$FINDFILID - STRUCTURE LEVEL 2
                           04B0  1160 ;
                           04B0  1161 ; STRUCTURE LEVEL 2
                           04B0  1162 ;
                           04B0  1163 FIND_LEVEL2:
         EF 34 A5   0D  E1 04B0  1164          BBC     #FH2$V_DIRECTORY,FH2$L_FILECHAR(R5),BADDIR ;DIRECTORY BIT MUST BE SE
                           04B5  1165
                           04B5  1166          ASSUME  FAT$B_RATTRIB EQ FAT$B_RTYPE+1
              0802 8F   B1 04B5  1167          CMPW    #<FAT$M_NOSPAN @ 8 + FAT$C_VARIABLE>,- ;VARIABLE LENGTH
                   64      04B9  1168                  FAT$B_RTYPE(R4)               ;RECORDS NOT CROSSING BLOCK BOUNDARIES
                   E8   12 04BA  1169          BNEQ    BADDIR                        ;BRANCH IF BAD RECORD ATTRIBUTES
                           04BC  1170 ;
                           04BC  1171 ; ***** NOTE THAT EACH BLOCK MUST END IN A RECORD SIZE OF -1
                           04BC  1172 ; ***** A RECORD IS NOT ALLOWED TO EXACTLY FILL THE BLOCK
                           04BC  1173 ; ***** PDP-11 FILE CONTROL SERVICES WILL READ THIS FILE CORRECTLY, BUT
                           04BC  1174 ; ***** WILL NOT WRITE IT PROPERLY.  LIKEWISE FOR RMS-11 AND RMS-32
                           04BC  1175 ;
                           04BC  1176 FIND_LEVEL2_1:
          58    08 BC   7D 04BC  1177          MOVQ    @FILDSC(AP),R8                ;R8 = SIZE, R9 = ADDRESS OF FILE NAME STRING
                   5A   D4 04C0  1178          CLRL    R10                           ;ASSUME DEFAULT VERSION
              53   58   7D 04C2  1179          MOVQ    R8,R3                         ;COPY FILE NAME DESCRIPTOR
          64  53   2E   3A 04C5  1180          LOCC    #^A/./,R3,(R4)                ;FIND FILE TYPE DELIMITER IF PRESENT
                   07   13 04C9  1181          BEQL    40$                           ;BRANCH IF NOT PRESENT
              53   70   9E 04CB  1182          MOVAB   -(R0),R3                      ;SIZE OF REMAINING STRING
            54  01 A1   9E 04CE  1183          MOVAB   1(R1),R4                      ;ADDRESS OF STRING BEYOND DELIMITER
          64  53   3B   3A 04D2  1184 40$:     LOCC    #^A/;/,R3,(R4)                ;SEE IF VERSION DELIMITER PRESENT
                   06   12 04D6  1185          BNEQ    60$                           ;BRANCH IF IT IS
          64  53   2E   3A 04D8  1186          LOCC    #^A/./,R3,(R4)                ;TRY ALTERNATE VERSION DELIMITER
                   34   13 04DC  1187          BEQL    120$                          ;BRANCH IF NO VERSION STRING PRESENT
                           04DE  1188 ;
                           04DE  1189 ; R0 = BYTE COUNT OF VERSION STRING PLUS DELIMITER
                           04DE  1190 ; R1 = ADDRESS OF VERSION DELIMITER
                           04DE  1191 ;
              58   50   C2 04DE  1192 60$:     SUBL    R0,R8                         ;REDUCE FILE NAME STRING SIZE
                           04E1  1193                                                ;ELIMINATING VERSION STRING AND DELIMITER
                   7E   DF 04E1  1194          PUSHAL  -(SP)                         ;RESERVE LONG WORD FOR VERSION NUMBER
                           04E3  1195                                                ;AND PUSH ITS ADDRESS
                01 A1   9F 04E3  1196          PUSHAB  1(R1)                         ;ADDRESS OF VERSION STRING
                   70   9F 04E6  1197          PUSHAB  -(R0)                         ;SIZE OF VERSION STRING
       00000000'EF 03   FB 04E8  1198          CALLS   #3,FIL$CVT_DTB                ;CONVERT DECIMAL VERSION STRING TO BINARY
                B8 50   E9 04EF  1199          BLBC    R0,BADFILNAM                  ;BRANCH IF SYNTAX ERROR IN VERSION STRING
              5A   8E   D0 04F2  1200          MOVL    (SP)+,R10                     ;FETCH EXPLICIT VERSION NUMBER
                           04F5  1201 ;
                           04F5  1202 ; R6  = ADDRESS OF LAST LBN READ FROM DIRECTORY FILE (FIRST - 1)
                           04F5  1203 ; R7  = ADDRESS OF LAST LBN (INCLUSIVE) TO BE READ FROM DIRECTORY FILE
                           04F5  1204 ; R8  = SIZE OF NAME STRING TO SCAN FOR
                           04F5  1205 ; R9  = ADDRESS OF NAME STRING TO SCAN FOR
                           04F5  1206 ; R10 = FILE VERSION NUMBER IF EXPLICIT, OR 0 IF DEFAULT TO LATEST VERSION
                           04F5  1207 ;
                   1B   11 04F5  1208          BRB     120$                          ;BEGIN LOOP AT BOTTOM
                           04F7  1209 ;
                           04F7  1210 ; R5 = ADDRESS OF NEXT RECORD
                           04F7  1211 ;
            54  05 A5   9A 04F7  1212 100$:    MOVZBL  DIR$B_NAMECOUNT(R5),R4     ;GET SIZE OF "NAME.TYP" STRING
06 A5   54  00  69 58   2D 04FB  1213          CMPC5   R8,(R9),#0,R4,DIR$T_NAME(R5) ;SEE IF STRINGS MATCH
                   1D   13 0502  1214          BEQL    200$                          ;BRANCH IF THEY DO
                   10   19 0504  1215          BLSS    140$                          ;BRANCH IF BEYOND WHERE NAME WOULD GO
              50    65    3C 0506           MOVZWL  DIR$W_SIZE(R5),R0           ;USING THE SIZE OF THIS RECORD
        55    02 A540    9E 0509           MOVAB   2(R5)[R0],R5                ;FORM ADDRESS OF NEXT RECORD
                  65    B5 050E  110$:    TSTW    DIR$W_SIZE(R5)              ;END OF BLOCK? (MARKED WITH -1)
                  E5    14 0510           BGTR    100$                        ;BRANCH IF NOT
        06 56     57    F3 0512  120$:    AOBLEQ  R7,R6,160$
        50   0910 8F    3C 0516  140$:    MOVZWL  #SS$_NOSUCHFILE,R0          ;CANNOT FIND FILE
                        04 051B  150$:    RET
                006D    30 051C  160$:    BSBW    READ_DIR_LBN                ;READ THE NEXT DIRECTORY LBN
                  ED    11 051F           BRB     110$
                           0521  ;
                           0521  ; FOUND A MATCH OF FILE NAME AND TYPE
                           0521  ;
                  54    D6 0521  200$:    INCL    R4                          ;ROUND UP NAME COUNT
              54  01    CA 0523           BICL    #1,R4                       ;TO EVEN NUMBER OF BYTES
        53    06 A544    9E 0526           MOVAB   DIR$T_NAME(R5)[R4],R3       ;ADDRESS OF FIRST VERSION ENTRY
              50    65    3C 052B           MOVZWL  DIR$W_SIZE(R5),R0           ;SIZE OF THIS RECORD
        55    02 A540    9E 052E           MOVAB   2(R5)[R0],R5                ;FORM ADDRESS OF BEGINNING OF NEX RECORD
                           0533                                               ;WHICH IS ALSO THE END OF THE VERSIONS
                  5A    D5 0533           TSTL    R10                         ;LATEST VERSION DESIRED?
                  11    13 0535           BEQL    240$                        ;BRANCH IF YES, R3 IS ADDRESS OF
                           0537                                               ;DESIRED VERSION AND FILE ID
              63  5A    B1 0537  230$:    CMPW    R10,DIR$W_VERSION(R3)       ;IS THIS THE RIGHT VERSION?
                  0C    13 053A           BEQL    240$                        ;BRANCH IF YES
                  D8    1A 053C           BGTRU   140$                        ;BRANCH IF PAST WHERE IT WOULD BE
              53  08    C0 053E           ADDL    #DIR$C_VERSION,R3           ;NEXT VERSION ENTRY
              55  53    D1 0541           CMPL    R3,R5                       ;END OF RECORD?
                  F1    1F 0544           BLSSU   230$                        ;BRANCH IF NOT, CHECK NEXT VERSION
                  C6    11 0546           BRB     110$                        ;VERSION NOT IN THIS VERSION CHAIN
                           0548                                               ;SEE IF IT IS IN THE NEXT RECORD
                           0548  ;
                           0548  ; FOUND THE FILE ID, RETURN IT TO CALLER
                           0548  ;
        56    02 A3     7D 0548  240$:    MOVQ    DIR$W_FID(R3),R6            ;GET THE FILE ID
              57    57    3C 054C           MOVZWL  R7,R7
  18 BC       02 A3  06  28 054F           MOVC3   #6,DIR$W_FID(R3),@FILID(AP) ;AND RETURN IT TO THE CALLER
                           0555  ;
                           0555  ; SEE IF WE SHOULD MAKE A CACHE ENTRY FOR THIS LOOKUP
                           0555  ; R6,R7 = FID
                           0555  ;
                           0555  EXIT_FILID_FND:
                  5B    D5 0555           TSTL    R11                         ;IS THE CACHE ENABLED?
                  2C    13 0557           BEQL    100$                        ;BRANCH IF NOT, JUST RETURN THE FID
              D6 AD     95 0559           TSTB    FIL$T_DIR_NAM+ENTRY(FP)     ;WAS LOOKUP FOR A DIRECTORY?
                  07    12 055C           BNEQ    20$                         ;BRANCH IF YES, MAKE A CACHE ENTRY
              F4 AD     D5 055E           TSTL    ENTRY_ADR(FP)               ;CACHE HIT FOR THIS DIR HDR?
                  22    12 0561           BNEQ    100$                        ;BRANCH IF YES
                  08    11 0563           BRB     30$                         ;MAKE THE CACHE ENTRY FOR THIS DIR HDR
        EE AD     56    D0 0565  20$:     MOVL    R6,FIL$A_DIR_OFID+ENTRY(FP)  ;STORE THE FID FOUND
        F2 AD     57    B0 0569           MOVW    R7,FIL$A_DIR_OFID+4+ENTRY(FP)
        58    08 AB     D0 056D  30$:     MOVL    FIL$L_DIRNXT(R11),R8        ;GET OFFSET TO FREE SPACE
     50   58    24    C1 0571           ADDL3   #FIL$C_DIR_SIZE,R8,R0       ;FORM OFFSET TO END OF NEW ENTRY
        0C AB     50    D1 0575           CMPL    R0,FIL$L_DIRMAX(R11)        ;ENOUGH SPACE FOR NEW ENTRY?
                  0A    14 0579           BGTR    90$                         ;BRANCH IF NOT
        08 AB     50    D0 057B           MOVL    R0,FIL$L_DIRNXT(R11)        ;YES, ALLOCATE THE NEW ENTRY
  6B48  D0 AD     24    28 057F           MOVC3   #FIL$C_DIR_SIZE,ENTRY(FP),(R11)[R8] ;AND WRITE IT
                           0585  90$:
              50    01    3C 0585  100$:    MOVZWL  #SS$_NORMAL,R0              ;INDICATE SUCCESSFUL COMPLETION
              04  0588  1273          RET
                  0589  1274  ;
                  0589  1275  ; BAD DIRECTORY FILE
                  0589  1276  ;
                  0589  1277  BADDIR1:
        FF18  31  0589  1278          BRW     BADDIR
```

```
                       058C   1280         .SBTTL  READ_DIR_LBN - READ NEXT DIRECTORY LBN
                       058C   1281 ;++
                       058C   1282 ; FUNCTIONAL DESCRIPTION:
                       058C   1283 ;
                       058C   1284 ;       READ THE NEXT DIRECTORY LBN FROM THE DISK OR POINT AT
                       058C   1285 ;       THE CACHED COPY IF ONE IS PRESENT
                       058C   1286 ;
                       058C   1287 ; CALLING SEQUENCE:
                       058C   1288 ;
                       058C   1289 ;       BSBW    READ_DIR_LBN
                       058C   1290 ;
                       058C   1291 ; INPUT:
                       058C   1292 ;
                       058C   1293 ;       R6 = DESIRED LBN
                       058C   1294 ;       DIRBUF(AP) = BUFFER ADDRESS TO READ IT INTO
                       058C   1295 ;       CHAN(AP) = CHANNEL FOR FIL$RDWRTLBN
                       058C   1296 ;       DIR_BFCNT(FP) = COUNT OF BUFFERS REMAINING IN DIR LBN CACHE
                       058C   1297 ;       DIR_BUF(FP) = ADDRESS OF NEXT BUFFER IN DIR LBN CACHE
                       058C   1298 ;
                       058C   1299 ; OUTPUTS:
                       058C   1300 ;
                       058C   1301 ;       R5 = ADDRESS OF DESIRED DIRECTORY LBN
                       058C   1302 ;       RSB IF SUCCESSFUL
                       058C   1303 ;       RET WITH STATUS IN R0 IF ERROR
                       058C   1304 ;       R0, R1 DESTROYED, OTHERS PRESERVED
                       058C   1305 ;
                       058C   1306 ;--
                       058C   1307
                       058C   1308 READ_DIR_LBN:
          FC AD    D5  058C   1309         TSTL    DIR_BFCNT(FP)                  ;ANYTHING LEFT IN DIR LBN CACHE?
             0E    13  058F   1310         BEQL    20$                            ;BRANCH IF NOT
          FC AD    D7  0591   1311         DECL    DIR_BFCNT(FP)                  ;COUNT ANOTHER BUFFER USED
     55   F8 AD    D0  0594   1312         MOVL    DIR_BUF(FP),R5                 ;LOAD ADDRESS OF BUFFER
F8 AD   0200 C5    DE  0598   1313         MOVAL   512(R5),DIR_BUF(FP)            ;AND POINT TO NEXT BUFFER IF ANY
                   05  059E   1314 10$:    RSB
                       059F   1315 ;
                       059F   1316 ; DIR LBN CACHE RAN OUT OF BLOCKS OR NEVER HAD ANY AT ALL
                       059F   1317 ;
     55   10 AC    D0  059F   1318 20$:    MOVL    DIRBUF(AP),R5                  ;ADDRESS OF BUFFER TO READ INTO
                       05A3   1319         READLBN CHAN(AP),R6,(R5)               ;READ THE DESIRED LBN
   7E   01   09    9C  05A3                ROTL    #9,#1,-(SP)
        7E   21    3C  05A7                MOVZWL  #IO$_READLBLK,-(SP)
             65    DF  05AA                PUSHAL  (R5)
             56    DD  05AC                PUSHL   R6
          04 AC    DD  05AE                PUSHL   CHAN(AP)
   0000'CF   05    FB  05B1                CALLS   #5,W^FIL$RDWRTLBN
          E5 50    E8  05B6   1320         BLBS    R0,10$                         ;BRANCH IF READ SUCCESSFULLY
                   04  05B9   1321         RET                                    ;RETURN ERROR STATUS
```

```
05BA            .SBTTL  RDCHKFILHDR - READ AND CHECK FILE HEADER
05BA      ;++
05BA      ; FUNCTIONAL DESCRIPTION:
05BA      ;
05BA      ;       RDCHKFILHDR READS AND VALIDATES A FILE HEADER GIVEN ITS FILE ID
05BA      ; AND THE INDEX FILE HEADER AS RETURNED BY FIL$MOUNT.
05BA      ;
05BA      ; CALLING SEQUENCE:
05BA      ;
05BA      ;       CALLG   ARGLIST,FIL$RDCHKFILHDR
05BA      ;
05BA      ; INPUT PARAMETERS:
05BA      ;
05BA      ;       CHAN(AP)                        CHANNEL ON WHICH DEVICE IS ASSIGNED
05BA      ;       UNUSED                          UNUSED PARAMETER
05BA      ;       IXFHDR(AP)                      ADR OF INDEX FILE HEADER AS RETURNED BY FIL$MOUNT
05BA      ;       FILHDR(AP)                      ADDRESS OF 512 BYTE BUFFER FOR FILE HEADER
05BA      ;       STATBLK(AP)                     ADR OF 2 LONG WORD AREA TO RETURN STATISTICS BLOCK
05BA      ;       FILID(AP)                       ADDRESS OF 3 WORD FILE ID OF DESIRED FILE HEADER
05BA      ;       RTRVPTRLEN(AP)  =               ADDRESS TO RETURN THE NUMBER OF
05BA      ;                                       BYTES OF RETRIEVAL POINTERS STORED
05BA      ;                                       ***** OPTIONAL PARAMETER *****
05BA      ;       RTRVPTRBUF(AP)  =               ADDRESS OF RETRIEVAL POINTER
05BA      ;                                       BUFFER DESCRIPTOR.  THIS PARAMETER
05BA      ;                                       IS PRESENT IF AND ONLY IF
05BA      ;                                       RTRVPTRLEN IS PRESENT.
05BA      ;                                       THE RETRIEVAL POINTERS ARE RETURNED IN
05BA      ;                                       THE FORM 32 BIT BLOCK COUNT, 32 BIT LBN
05BA      ;                                       A ZERO BUFFER DESCRIPTOR ADDRESS OR A
05BA      ;                                       ZERO BUFFER ADDRESS MEANS DON'T
05BA      ;                                       RETURN RETRIEVAL POINTER INFO
05BA      ;
05BA      ; IMPLICIT INPUTS:
05BA      ;
05BA      ;       NONE
05BA      ;
05BA      ; OUTPUT PARAMETERS:
05BA      ;
05BA      ;       R0 = SYSTEM STATUS CODE
05BA      ;
05BA      ; IMPLICIT OUTPUTS:
05BA      ;
05BA      ;       NONE
05BA      ;
05BA      ; COMPLETION CODES:
05BA      ;
05BA      ;       SS$_NORMAL                      SUCCESSFUL COMPLETION
05BA      ;
05BA      ; SIDE EFFECTS:
05BA      ;
05BA      ;       NONE
05BA      ;
05BA      ;
05BA      ; EQUATED SYMBOLS
05BA      ;
05BA      ;    OFFSETS FROM AP
05BA      ;
```

```
                   00000000  05BA  1380            ARGCNT          =       0
                   00000004  05BA  1381            CHAN            =       4
                   0000000C  05BA  1382            IXFHDR          =       12
                   00000010  05BA  1383            FILHDR          =       16
                   00000014  05BA  1384            STATBLK         =       20
                   00000018  05BA  1385            FILID           =       24
                   0000001C  05BA  1386            RTRVPTRLEN      =       28      ; OPTIONAL PARAMETER
                   00000020  05BA  1387            RTRVPTRBUF      =       32      ; PRESENT IF AND ONLY IF RTRVPTRLEN IS
                             05BA  1388 ;
                             05BA  1389 ;      OFFSETS FROM FP
                             05BA  1390 ;
                             05BA  1391            $OFFSET 0,NEGATIVE,<-
                             05BA  1392            <HDRCNT>,-                      ;COUNT OF FILE HEADERS READ
                             05BA  1393            <TMPRTRVLEN>,-                  ;TEMP RTRV PTR BYTE COUNT
                             05BA  1394            <TMPRTRVDSC,8>-                 ;TEMP RTRV PTR BUFFER DESCRIPTOR
                             05BA  1395            >
                             FFFC        HDRCNT:
                             FFF8        TMPRTRVLEN:
                             FFF0        TMPRTRVDSC:
                             05BA  1396 ;
                             05BA  1397 ;--
                             05BA  1398
                             05BA  1399 FIL$RDCHKFILHDR::
                       007C  05BA  1400            .WORD   ^M<R2,R3,R4,R5,R6>
          7E    01      CE   05BC  1401            MNEGL   #1,-(SP)                ;COUNT OF HEADER BLOCKS READ
                7E      D4   05BF  1402            CLRL    -(SP)                   ;INIT RTRV PTR BYTE COUNT
                7E      7C   05C1  1403            CLRQ    -(SP)                   ;ASSUME NO RTRV PTR BUFFER
          08    6C      D1   05C3  1404            CMPL    ARGCNT(AP),#RTRVPTRBUF/4  ;RTRV PTR PARAMS PRESENT?
                0D      19   05C6  1405            BLSS    2$                      ;BRANCH IF NOT
       50    20 AC      D0   05C8  1406            MOVL    RTRVPTRBUF(AP),R0       ;RTRV BUFFER DESCRIPTOR ADDRESS
                07      13   05CC  1407            BEQL    2$                      ;BRANCH IF NONE SPECIFIED
       F0 AD    60      7D   05CE  1408            MOVQ    (R0),TMPRTRVDSC(FP)     ;MAKE COPY OF BUF DESCRIPTOR
          1C BC         D4   05D2  1409            CLRL    @RTRVPTRLEN(AP)         ;INIT RETURNED BYTE COUNT
                             05D5  1410
                             05D5  1411            ASSUME  FILHDR EQ IXFHDR+4
       52    0C AC      7D   05D5  1412 2$:        MOVQ    IXFHDR(AP),R2           ;R2 = INDEX FILE HEADER ADDRESS
                             05D9  1413                                            ;R3 = FILE HEADER ADDRESS
                             05D9  1414            ASSUME  FILID EQ STATBLK+4
       54    14 AC      7D   05D9  1415            MOVQ    STATBLK(AP),R4          ;R4 = RETURN STATBLK ADDRESS
                             05DD  1416                                            ;R5 = ADDRESS OF FILE ID
          7E    65      7D   05DD  1417            MOVQ    (R5),-(SP)              ;COPY FILE ID TO WRITABLE SCRATCH
          55    5E      D0   05E0  1418            MOVL    SP,R5                   ;AND REMEMBER ITS ADDRESS
          56    7E      7E   05E3  1419            MOVAQ   -(SP),R6                ;RESERVE SCRATCH STATISTICS BLOCK
                             05E6  1420                                            ;AND SAVE ITS ADDRESS IN R6
                64      7C   05E6  1421            CLRQ    (R4)                    ;INIT THE RETURN STAT BLOCK
                64      D7   05E8  1422            DECL    (R4)                    ;-1 LBN MEANS NOT YET STORED
          7E    65      3C   05EA  1423 5$:        MOVZWL  (R5),-(SP)              ;FILE NUMBER TO LONG WORD
 05 06 A2       0A      E1   05ED  1424            BBC     #FH2$V_BIGFILNUM,FH2$W_STRUCLEV(R2),10$ ;BRANCH IF NO BIG FIL NUM
 02 AE       05 A5      90   05F2  1425            MOVB    FID$B_NMX(R5),2(SP)     ;HIGH 8 BITS OF RVN COMPLETE FILE NUMBER
          50    8E      D0   05F7  1426 10$:       MOVL    (SP)+,R0                ;IF FILE ID IS ZERO,
                03      12   05FA  1427            BNEQ    12$
              006D      31   05FC  1428            BRW     40$                     ;THEN READ LAST HEADER BLOCK
   51    01FE C2        3C   05FF  1429 12$:       MOVZWL  FH2$W_VBNOFFSET(R2),R1  ;RECOVER VBN OFFSET FROM INDEX FILE HEADER
          50    51      C0   0604  1430            ADDL    R1,R0                   ;ADD VBN OFFSET TO FORM INDEX FILE VBN
             FC AD      D6   0607  1431            INCL    HDRCNT(FP)              ;COUNT EACH HEADER READ
                             060A  1432            READVBN CHAN(AP),R0,(R3),(R2)   ;READ THE FILE HEADER
                62      DF   060A                          PUSHAL  (R2)
                63  DF  060C                 PUSHAL  (R3)
                50  DD  060E                 PUSHL   R0
             04 AC  DD  0610                 PUSHL   CHAN(AP)
      0685'CF   04  FB  0613                 CALLS   #4,W^FIL$READVBN
             62 50  E9  0618  1433           BLBC    R0,50$                  ;BRANCH IF ERROR
      5D FC AD  1F  E0  061B  1434           BBS     #31,HDRCNT(FP),50$      ;BRANCH IF JUST RE-READING MAIN HEADER
             50 55  D0  0620  1435           MOVL    R5,R0                   ;GET FILE ID ADDRESS
             51 53  D0  0623  1436           MOVL    R3,R1                   ;ADDRESS OF FILE HEADER
              0153  30  0626  1437           BSBW    FIL$CHKFILHDR           ;CHECK THE FILE HEADER
       52    0C AC  D0  0629  1438           MOVL    IXFHDR(AP),R2           ;INDEX FILE HEADER ADDRESS
             F0 AD  DF  062D  1439           PUSHAL  TMPRTRVDSC(FP)          ;RTRV PTR BUF DESCRIPTOR
             F8 AD  DF  0630  1440           PUSHAL  TMPRTRVLEN(FP)          ;ADDRESS TO RETURN BYTE COUNT
                56  DD  0633  1441           PUSHL   R6                      ;ADDRESS OF SCRATCH STAT BLOCK
                53  DD  0635  1442           PUSHL   R3                      ;ADDRESS OF FILE HEADER
      0718'CF   04  FB  0637  1443           CALLS   #4,W^FIL$STATBLK        ;READ STATISTICS BLOCK
       51    F8 AD  D0  063C  1444           MOVL    TMPRTRVLEN(FP),R1       ;ANY RTRV PTR INFO TO RETURN?
                16  13  0640  1445           BEQL    16$                     ;ZERO IF NONE REQUESTED
         1C BC  51  C0  0642  1446           ADDL    R1,@RTRVPTRLEN(AP)      ;ACCUMULATE RTRVPTR BYTE COUNT
         F0 AD  51  D1  0646  1447           CMPL    R1,TMPRTRVDSC(FP)       ;MORE SPACE NEEDED THAN WOULD FIT?
                04  15  064A  1448           BLEQ    14$                     ;BRANCH IF NOT
       51    F0 AD  D0  064C  1449           MOVL    TMPRTRVDSC(FP),R1       ;SAY WE USED IT ALL UP
         F4 AD  51  C0  0650  1450 14$:      ADDL    R1,TMPRTRVDSC+4(FP)     ;GET NEW STARTING ADDRESS
         F0 AD  51  C2  0654  1451           SUBL    R1,TMPRTRVDSC(FP)       ;AND CALC NEW SIZE REMAINING
             51 64  D2  0658  1452 16$:      MCOML   (R4),R1                 ;SEE IF START LBN HAS BEEN SET
                03  12  065B  1453           BNEQ    20$                     ;BRANCH IF IT HAS
             64 66  D0  065D  1454           MOVL    (R6),(R4)               ;SET IT ONCE ONLY
      04 A4  04 A6  C0  0660  1455 20$:      ADDL    4(R6),4(R4)             ;ADD IN THE SIZE FROM THIS HEADER
         65  0E A3  7D  0665  1456           MOVQ    FH2$W_EXT_FID(R3),(R5)  ;GET EXTENSION FILE ID IF ANY
              FF7E  31  0669  1457 30$:      BRW     5$                      ;READ THIS HEADER IF ANY
                        066C  1458 ;
                        066C  1459 ; LAST FILE HEADER READ, SEE IF MUST RE-READ THE ORIGINAL HEADER
                        066C  1460 ;
             FC AD  D5  066C  1461 40$:      TSTL    HDRCNT(FP)              ;WAS -1, BUMPED ONCE PER READVBN
                09  15  066F  1462           BLEQ    45$                     ;BRANCH IF STILL HAVE MASTER FILE HEADER
         65  18 BC  7D  0671  1463           MOVQ    @FILID(AP),(R5)         ;ORIGINAL FILE ID AGAIN
      EF FC AD  1F  E3  0675  1464           BBCS    #31,HDRCNT(FP),30$      ;SET SIGN BIT AND GO READ ORIGINAL HEADER
             50 01  3C  067A  1465 45$:      MOVZWL  #SS$_NORMAL,R0          ;RETURN SUCCESS STATUS
                04      067D  1466 50$:      RET
```

```
          067E  1468          .SBTTL  READVBN, WRITEVBN - READ/WRITE VIRTUAL BLOCK
          067E  1469  ;++
          067E  1470  ; FUNCTIONAL DESCRIPTION:
          067E  1471  ;
          067E  1472  ;       THESE ROUTINES READ OR WRITE A VIRTUAL BLOCK FROM A FILE.
          067E  1473  ; VOLUME IS SPECIFIED BY THE CHANNEL TO WHICH IT IS ASSIGNED, AND THE
          067E  1474  ; FILE IS SPECIFIED BY THE ADDRESS OF ITS FILE HEADER WHICH WAS PREVIOUSLY
          067E  1475  ; READ BY A CALL TO FIL$RDFILHDR.
          067E  1476  ;
          067E  1477  ; CALLING SEQUENCE:
          067E  1478  ;
          067E  1479  ;       CALLG   ARGLIST,FIL$READVBN
          067E  1480  ;       CALLG   ARGLIST,FIL$WRITEVBN
          067E  1481  ;
          067E  1482  ; INPUT PARAMETERS:
          067E  1483  ;
          067E  1484  ;       CHAN(AP)        =                   ;CHANNEL TO WHICH VOLUME IS ASSIGNED
          067E  1485  ;       VBN(AP)         =                   ;DESIRED VIRTUAL BLOCK NUMBER
          067E  1486  ;       BUFADR(AP)      =                   ;ADDRESS OF BUFFER TO READ INTO
          067E  1487  ;       FILHDR(AP)      =                   ;ADDRESS OF FILE HEADER
          067E  1488  ;
          067E  1489  ; IMPLICIT INPUTS:
          067E  1490  ;
          067E  1491  ;       NONE
          067E  1492  ;
          067E  1493  ; OUTPUT PARAMETERS:
          067E  1494  ;
          067E  1495  ;       R0 = SYSTEM STATUS CODE
          067E  1496  ;
          067E  1497  ; IMPLICIT OUTPUTS:
          067E  1498  ;
          067E  1499  ;       NONE
          067E  1500  ;
          067E  1501  ; COMPLETION CODES:
          067E  1502  ;
          067E  1503  ;       SS$_NORMAL                          SUCCESSFUL RETURN
          067E  1504  ;       SS$_ENDOFFILE                       SPECIFIED VBN BEYOND END OF FILE
          067E  1505  ;
          067E  1506  ; SIDE EFFECTS:
          067E  1507  ;
          067E  1508  ;       NONE
          067E  1509  ;
          067E  1510  ;
          067E  1511  ; EQUATED SYMBOLS:
          067E  1512  ;
          067E  1513  ;    OFFSET FROM AP
          067E  1514  ;
00000004  067E  1515          CHAN            =       4       ;CHANNEL TO WHICH VOLUME IS ASSIGNED
00000008  067E  1516          VBN             =       8       ;VIRTUAL BLOCK NUMBER
0000000C  067E  1517          BUFADR          =       12      ;BUFFER ADDRESS TO READ INTO
00000010  067E  1518          FILHDR          =       16      ;ADDRESS OF FILE HEADER
          067E  1519  ;
          067E  1520  ;    OFFSETS FROM FP
          067E  1521  ;
FFFFFFFC  067E  1522          IOFUNCTION      =       -4      ;SAVED I/O FUNCTION CODE
          067E  1523  ;
          067E  1524  ;--
```

```
                              067E   1525
                              067E   1526 FIL$WRITEVBN::
                        003C  067E   1527           .WORD    ^M<R2,R3,R4,R5>
             7E    20     3C  0680   1528           MOVZWL   #IO$_WRITELBLK,-(SP)
                   05     11  0683   1529           BRB      RDWRTVBN
                              0685   1530
                              0685   1531 FIL$READVBN::
                        003C  0685   1532           .WORD    ^M<R2,R3,R4,R5>
             7E    21     3C  0687   1533           MOVZWL   #IO$_READLBLK,-(SP)
                              068A   1534
                              068A   1535 RDWRTVBN:
       55      10 AC      D0  068A   1536           MOVL     FILHDR(AP),R5               ;BASE ADR OF FILE HEADER
                   31     10  068E   1537           BSBB     INIRTRVPTRSCAN              ;SET UP TO SCAN RETRIEVAL POINTERS
                              0690   1538 ;
                              0690   1539 ; R4 = POINTER TO FIRST RETRIEVAL POINTER,
                              0690   1540 ; R5 = POINTER TO FIRST BYTE BEYOND LAST RETRIEVAL POINTER
                              0690   1541 ; LOOP THROUGH RETRIEVAL POINTERS TO FIND THE ONE WHICH CONTAINS THE DESIRED VBN
                              0690   1542 ;
  53      08 AC    01     C3  0690   1543           SUBL3    #1,VBN(AP),R3               ;VBN BASE 0 TO LOOK FOR
                   3B     10  0695   1544 20$:      BSBB     GETRTRVPTR                  ;FETCH NEXT RETRIEVAL POINTER
             50    53     D1  0697   1545           CMPL     R3,R0                       ;IS VBN IN THIS RETRIEVAL POINTER
                   0E     19  069A   1546           BLSS     40$                         ;BRANCH IF YES
             53    50     C2  069C   1547           SUBL     R0,R3                       ;PASS OVER THAT MANY VBN'S
             55    54     D1  069F   1548           CMPL     R4,R5                       ;ANY MORE RETRIEVAL POINTERS?
                   F1     1F  06A2   1549           BLSSU    20$                         ;BRANCH IF YES
       50    0870 8F      3C  06A4   1550           MOVZWL   #SS$_ENDOFFILE,R0           ;RETURN END OF FILE INDICATION
                          04  06A9   1551           RET
                              06AA   1552 ;
                              06AA   1553 ; VBN IS IN THIS RETRIEVAL POINTER, R1 = STARTING LBN
                              06AA   1554 ;
       7E    01    09     9C  06AA   1555 40$:      ROTL     #9,#1,-(SP)                 ;NUMBER OF BYTES TO READ/WRITE
                FC AD     DD  06AE   1556           PUSHL    IOFUNCTION(FP)              ;FUNCTION CODE
                0C AC     DD  06B1   1557           PUSHL    BUFADR(AP)                  ;BUFFER TO TRANSFER TO/FROM
       7E    51    53     C1  06B4   1558           ADDL3    R3,R1,-(SP)                 ;LBN
                04 AC     DD  06B8   1559           PUSHL    CHAN(AP)                    ;CHANNEL
   0000'CF         05     FB  06BB   1560           CALLS    #5,W^FIL$RDWRTLBN           ;TRANSFER THE BLOCK
                          04  06C0   1561           RET
```

```
                    06C1  1563           .SBTTL  INIRTRVPTRSCAN - INITIALIZE RETRIEVAL POINTER SCAN
                    06C1  1564  ;++
                    06C1  1565  ; FUNCTIONAL DESCRIPTION:
                    06C1  1566  ;
                    06C1  1567  ;        LOCATE START AND END OF RETRIEVAL POINTERS IN A FILE HEADER.
                    06C1  1568  ;
                    06C1  1569  ; CALLING SEQUENCE:
                    06C1  1570  ;
                    06C1  1571  ;        BSBW    INIRTRVPTRSCAN
                    06C1  1572  ;
                    06C1  1573  ; INPUT:
                    06C1  1574  ;
                    06C1  1575  ;        R5 = FILE HEADER ADDRESS
                    06C1  1576  ;
                    06C1  1577  ; OUTPUT:
                    06C1  1578  ;
                    06C1  1579  ;        R4 = ADDRESS OF 1ST RETRIEVAL POINTER
                    06C1  1580  ;        R5 = ADDRESS OF FIRST BYTE BEYOND LAST RETREIVAL POINTER
                    06C1  1581  ;
                    06C1  1582  ;--
                    06C1  1583
                    06C1  1584  INIRTRVPTRSCAN:
 50   01 A5    9A   06C1  1585           MOVZBL  FH2$B_MPOFFSET(R5),R0     ;WORD OFFSET TO MAP AREA
  54   6540    3E   06C5  1586           MOVAW   (R5)[R0],R4               ;BASE ADR OF MAP AREA
 55   3A A5    9A   06C9  1587           MOVZBL  FH2$B_MAP_INUSE(R5),R5    ;NO. OF WORDS OF RTRV PTRS IN USE
  55   6445    3E   06CD  1588  10$:     MOVAW   (R4)[R5],R5               ;ADR JUST BEYOND LAST VALID RTRV PTR
               05   06D1  1589           RSB
```

```
                      06D2  1591          .SBTTL  GETRTRVPTR - CONVERT NEXT RETRIEVAL POINTER
                      06D2  1592 ;++
                      06D2  1593 ; FUNCTIONAL DESCRIPTION:
                      06D2  1594 ;
                      06D2  1595 ;         CONVERT NEXT RETRIEVAL POINTER TO NUMBER OF BLOCKS COVERED BY
                      06D2  1596 ;         POINTER AND STARTING LBN.
                      06D2  1597 ;
                      06D2  1598 ; CALLING SEQUENCE:
                      06D2  1599 ;
                      06D2  1600 ;         BSBW    GETRTRVPTR
                      06D2  1601 ;
                      06D2  1602 ; INPUTS:
                      06D2  1603 ;
                      06D2  1604 ;         R4 = ADDRESS OF NEXT RETRIEVAL POINTER
                      06D2  1605 ;
                      06D2  1606 ; OUTPUTS:
                      06D2  1607 ;
                      06D2  1608 ;         R0 = NUMBER OF BLOCKS COVERED BY THE RETRIEVAL POINTER
                      06D2  1609 ;         R1 = STARTING LOGICAL BLOCK NUMBER
                      06D2  1610 ;         R2,R3 PRESERVED
                      06D2  1611 ;
                      06D2  1612 ;--
                      06D2  1613
                      06D2  1614 GETRTRVPTR:
                      06D2  1615 ;
                      06D2  1616 ; STRUCTURE LEVEL 2 RETRIEVAL POINTERS
                      06D2  1617 ;         BITS 14:15 = RETRIEVAL POINTER FORMAT
                      06D2  1618 ;
50   64   02   0E  EF 06D2  1619 20$:     EXTZV   #FM2$V_FORMAT,#FM2$S_FORMAT,(R4),R0 ;FORMAT TO R0
                      06D7  1620          CASE    R0,<-
                      06D7  1621                  PLACEMENT,-             ;PLACEMENT FORMAT
                      06D7  1622                  FORMAT1,-               ;FORMAT 1
                      06D7  1623                  FORMAT2-                ;FORMAT 2
                      06D7  1624                  >
                      06E1  1625 ;
                      06E1  1626 ; FORMAT 3 = 8 BYTES
                      06E1  1627 ;
                      06E1  1628 ;         BITS 0:13 = BITS 16:29 OF COUNT - 1
                      06E1  1629 ;         BITS 14:15 = FORMAT = 3
                      06E1  1630 ;         BYTES 2-3 = BITS 0:15 OF COUNT - 1
                      06E1  1631 ;         BYTES 4-7 = LOGICAL BLOCK NUMBER
                      06E1  1632 ;
                      06E1  1633 FORMAT3:
     50   84   10  9C 06E1  1634          ROTL    #16,(R4)+,R0            ;FORM COUNT - 1
50   02   1E   00  F0 06E5  1635          INSV    #0,#30,#2,R0            ;ZERO HIGH 2 BITS
          51   84  D0 06EA  1636          MOVL    (R4)+,R1                ;GET LBN
               19  11 06ED  1637          BRB     INCRSB                  ;INCREMENT COUNT AND EXIT
                      06EF  1638 ;
                      06EF  1639 ; PLACEMENT CONTROL - THIS IS NOT A RETRIEVAL POINTER, RATHER IT
                      06EF  1640 ;         CONSISTS OF 2 BYTES OF PLACEMENT INFORMATION.  TREAT AS IF 0
                      06EF  1641 ;         LENGTH RETRIEVAL POINTER.
                      06EF  1642 ;         R0 = 0
                      06EF  1643 ;
                      06EF  1644 PLACEMENT:
          51   01  CE 06EF  1645          MNEGL   #1,R1                   ;IMPOSSIBLE LBN
          54   02  C0 06F2  1646          ADDL    #2,R4                   ;BUMP THE POINTER
               50  D4 06F5  1647          CLRL    R0                      ;CLEAR BLOCK COUNT
```

```
                        05  06F7              RSB
                            06F8      ;
                            06F8      ; FORMAT 1 = 4 BYTES
                            06F8      ;       BITS 0:7 = COUNT - 1
                            06F8      ;       BITS 8:13 = BITS 16:21 OF LOGICAL BLOCK NUMBER
                            06F8      ;       BYTES 2-3 = BITS 0:15  OF LOGICAL BLOCK NUMBER
                            06F8      ;
                            06F8      FORMAT1:
              50    84  D0  06F8              MOVL    (R4)+,R0                    ;FETCH ENTIRE RETRIEVAL POINTER
  51    50    06    08  EF  06FB              EXTZV   #FM2$V_HIGHLBN,#FM2$S_HIGHLBN,R0,R1 ;FETCH HIGH LBN BITS
        50    50    10  79  0700              ASHQ    #16,R0,R0                   ;FORM R1 = LBN
           50    FC A4  9A  0704              MOVZBL  -4(R4),R0                   ;REFETCH COUNT - 1
                            0708      INCRSB:
                    50  D6  0708              INCL    R0                          ;FORM COUNT
                        05  070A              RSB                                 ;AND RETURN
                            070B      ;
                            070B      ; FORMAT 2 = 6 BYTES
                            070B      ;
                            070B      ;       BITS 0:13 = COUNT - 1
                            070B      ;       BITS 14:15 = FORMAT = 2
                            070B      ;       BYTES 2-5 = LBN
                            070B      ;
                            070B      FORMAT2:
              50    84  3C  070B              MOVZWL  (R4)+,R0                    ;FETCH COUNT - 1 AND FORMAT BITS
  50    50    0E    00  EF  070E              EXTZV   #FM2$V_COUNT2,#FM2$S_COUNT2,R0,R0 ;COUNT - 1
              51    84  D0  0713              MOVL    (R4)+,R1                    ;LBN
                    F0  11  0716              BRB     INCRSB                      ;INCREMENT COUNT AND RETURN
```

```
0718          .SBTTL  STATBLK - GET FILE STATISTICS BLOCK
0718  ;++
0718  ; FUNCTIONAL DESCRIPTION:
0718  ;
0718  ;       GIVEN A FILE HEADER, RETURN THE FILE STATISTICS BLOCK
0718  ;       AND OPTIONALLY RETURN THE RETRIEVAL POINTERS
0718  ;
0718  ; CALLING SEQUENCE:
0718  ;
0718  ;       CALLG   ARGLIST,FIL$STATBLK
0718  ;
0718  ; INPUT PARAMETERS:
0718  ;
0718  ;       FILHDR(AP)      =               ;ADDRESS OF THE FILE HEADER
0718  ;       STATBLK(AP)     =               ;ADDRESS TO RETURN STATISTICS BLOCK
0718  ;       RTRVPTRLEN(AP)  =               ;ADDRESS TO RETURN THE NUMBER OF
0718  ;                                       ;BYTES OF RETRIEVAL POINTERS
0718  ;                                       ;FOUND IN THE FILE HEADER(S).
0718  ;                                       ;***** OPTIONAL PARAMETER *****
0718  ;       RTRVPTRBUF(AP)  =               ;ADDRESS OF RETRIEVAL POINTER
0718  ;                                       ;BUFFER DESCRIPTOR.  THIS PARAMETER
0718  ;                                       ;IS PRESENT IF AND ONLY IF
0718  ;                                       ;RTRVPTRLEN IS PRESENT.
0718  ;                                       ;ZERO DESCRIPTOR ADDRESS OR ZERO
0718  ;                                       ;BUFFER ADDRESS MEANS DON'T
0718  ;                                       ;RETURN RETRIEVAL POINTER INFO
0718  ;
0718  ; IMPLICIT INPUTS:
0718  ;
0718  ;       NONE
0718  ;
0718  ; OUTPUT PARAMETERS:
0718  ;
0718  ;       R0 = SYSTEM STATUS CODE
0718  ;       STATBLK CONTAINS 2 LONGWORDS
0718  ;               LBN OF 1ST BLOCK IF CONTIGUOUS OR ZERO IF NOT
0718  ;               SIZE OF FILE IN BLOCKS
0718  ;       RTRVPTRLEN RECEIVES THE NUMBER OF BYTES OF RETRIEVAL POINTER
0718  ;               INFORMATION THAT WOULD HAVE BEEN STORED IN THE RETRIEVAL
0718  ;               POINTER BUFFER GIVEN A LARGE ENOUGH BUFFER.
0718  ;       THE RETRIEVAL POINTER BUFFER RECEIVES NORMALIZED RETRIEVAL
0718  ;               POINTERS IN THE FORMAT 32 BIT COUNT, 32 BIT STARTING LBN
0718  ;
0718  ; IMPLICIT OUTPUTS:
0718  ;
0718  ;       NONE
0718  ;
0718  ; COMPLETION CODES:
0718  ;
0718  ;       SS$_NORMAL                      SUCCESSFUL COMPLETION
0718  ;
0718  ; SIDE EFFECTS:
0718  ;
0718  ;       NONE
0718  ;
0718  ;
0718  ; EQUATED SYMBOLS:
```

```
                          0718  1733 ;
                          0718  1734 ;     OFFSETS FROM AP
                          0718  1735 ;
               00000000   0718  1736          ARGCNT          =       0       ;NUMBER OF ARGUMENTS
               00000004   0718  1737          FILHDR          =       4       ;ADDRESS OF FILE HEADER
               00000008   0718  1738          STATBLK         =       8       ;ADDRESS TO RETURN STATISTICS BLOCK
               0000000C   0718  1739          RTRVPTRLEN      =       12      ;ADDRESS TO RETURN COUNT OF BYTES
                          0718  1740                                          ;STORED IN THE RETRIEVAL POINTER BUFFER
               00000010   0718  1741          RTRVPTRBUF      =       16      ;ADDRESS OF RETRIEVAL POINTER
                          0718  1742                                          ;BUFFER DESCRIPTOR
                          0718  1743 ;
                          0718  1744 ;--
                          0718  1745
                          0718  1746 FIL$STATBLK::
                   00FC   0718  1747          .WORD   ^M<R2,R3,R4,R5,R6,R7>
                 56   7C  071A  1748          CLRQ    R6                      ;ASSUME NOT DOING RETRIEVAL POINTERS
             04  6C   D1  071C  1749          CMPL    ARGCNT(AP),#RTRVPTRBUF/4 ;RTRV PTR PARAMS PRESENT?
                 0F   19  071F  1750          BLSS    5$                      ;BRANCH IF NOT
         50   10 AC   D0  0721  1751          MOVL    RTRVPTRBUF(AP),R0       ;ADDRESS OF BUFFER DESCRIPTOR
                 09   13  0725  1752          BEQL    5$                      ;BRANCH IF NOT SPECIFIED
             56  60   7D  0727  1753          MOVQ    (R0),R6                 ;R6 = MAX SIZE, R7 = BUFFER ADR
             56  07   CA  072A  1754          BICL    #7,R6                   ;EVEN MULTIPLE OF 8 BYTES
              0C BC   D4  072D  1755          CLRL    @RTRVPTRLEN(AP)         ;INIT RETURN BYTE COUNT
         55   04 AC   D0  0730  1756 5$:      MOVL    FILHDR(AP),R5           ;ADDRESS OF FILE HEADER
         50   34 A5   9A  0734  1757          MOVZBL  FH2$L_FILECHAR(R5),R0   ;FILE CHARACTERISTICS IF LEVEL 2
 7E   50   01    07   EF  0738  1758 10$:     EXTZV   #FH2$V_CONTIG,#1,R0,-(SP) ;CONTIGUOUS BIT TO TOP OF STACK
               FF81   30  073D  1759          BSBW    INIRTRVPTRSCAN          ;INIT FOR SCAN OF RETRIEVAL POINTERS
                 53   D4  0740  1760          CLRL    R3                      ;INIT REGISTER TO COUNT BLOCKS
                 29   11  0742  1761          BRB     50$                     ;START AT BOTTOM OF LOOP IN CASE
                          0744  1762                                          ;FILE HAS NO RETRIEVAL POINTERS
               FF8B   30  0744  1763 20$:     BSBW    GETRTRVPTR              ;GET THE NEXT RETRIEVAL POINTER
                 53   D5  0747  1764          TSTL    R3                      ;IS THIS FIRST RTRV PTR?
                 0A   12  0749  1765          BNEQ    40$                     ;BRANCH IF ALREADY COUNTED SOME
                          074B  1766 ;
                          074B  1767 ; FIRST RETRIEVAL POINTER
                          074B  1768 ;
                 50   D5  074B  1769          TSTL    R0                      ;IGNORE EMPTY ONES
                 1E   13  074D  1770          BEQL    50$
              03 6E   E9  074F  1771          BLBC    (SP),40$                ;BRANCH IF FILE NOT CONTIGUOUS
                          0752  1772                                          ;0(SP) = 0 IN THIS CASE
             6E  51   D0  0752  1773          MOVL    R1,(SP)                 ;SET LBN OF 1ST NON-ZERO RTRV PTR
             53  50   C0  0755  1774 40$:     ADDL    R0,R3                   ;ACCUMULATE COUNT OF BLOCKS
                 56   D5  0758  1775          TSTL    R6                      ;ANY MORE ROOM FOR RTRV PTRS?
                 09   13  075A  1776          BEQL    45$                     ;BRANCH IF NO MORE BUFFER SPACE
             87  50   D0  075C  1777          MOVL    R0,(R7)+                ;STORE SIZE OF RETRIEVAL POINTER
             87  51   D0  075F  1778          MOVL    R1,(R7)+                ;AND STORE LBN
             56  08   C2  0762  1779          SUBL    #8,R6                   ;USED 8 MORE BYTES OF SPACE
                 57   D5  0765  1780 45$:     TSTL    R7                      ;DOES CALLER WANT RTRV PTR INFO?
                 04   13  0767  1781          BEQL    50$                     ;BRANCH IF NOT, DON'T COUNT POINTERS
          0C BC  08   C0  0769  1782          ADDL    #8,@RTRVPTRLEN(AP)      ;UPDATE RTRV PTR BYTE COUNT
             55  54   D1  076D  1783 50$:     CMPL    R4,R5                   ;ANY MORE RETRIEVAL POINTERS?
                 D2   1F  0770  1784          BLSSU   20$                     ;BRANCH IF YES
                 04   BA  0772  1785          POPR    #^M<R2>                 ;GET SAVED STARTING LBN
          08 BC  52   7D  0774  1786          MOVQ    R2,@STATBLK(AP)         ;RETURN THE STATISTICS BLOCK
             50  01   3C  0778  1787          MOVZWL  #SS$_NORMAL,R0          ;SUCCESSFUL COMPLETION
                      04  077B  1788          RET
```

```
                          077C  1790          .SBTTL  FIL$CHKFILHDR - CHECK FILE HEADER VALIDITY
                          077C  1791 ;++
                          077C  1792 ; FUNCTIONAL DESCRIPTION:
                          077C  1793 ;
                          077C  1794 ;        CHECK THE VALIDITY OF A FILE HEADER
                          077C  1795 ;
                          077C  1796 ; CALLING SEQUENCE:
                          077C  1797 ;
                          077C  1798 ;        BSBW    FIL$CHKFILHDR
                          077C  1799 ;
                          077C  1800 ; INPUT PARAMETERS:
                          077C  1801 ;
                          077C  1802 ;        R0 = ADDRESS OF FILE ID
                          077C  1803 ;        R1 = ADDRESS OF FILE HEADER
                          077C  1804 ;
                          077C  1805 ; IMPLICIT INPUTS:
                          077C  1806 ;
                          077C  1807 ;        NONE
                          077C  1808 ;
                          077C  1809 ; OUTPUT PARAMETERS:
                          077C  1810 ;
                          077C  1811 ;        RSB TO CALLER IF FILE HEADER VALID
                          077C  1812 ;        RET IF NOT VALID WITH R0 = ERROR STATUS
                          077C  1813 ;
                          077C  1814 ; IMPLICIT OUTPUTS:
                          077C  1815 ;
                          077C  1816 ;        NONE
                          077C  1817 ;
                          077C  1818 ; COMPLETION CODES:
                          077C  1819 ;
                          077C  1820 ;        SS$_BADFILEHDR          FILE ID CODES DON'T MATCH
                          077C  1821 ;        SS$_NOSUCHFILE          FILE IS MARKED AS DELETED
                          077C  1822 ;
                          077C  1823 ; SIDE EFFECTS:
                          077C  1824 ;
                          077C  1825 ;        NONE
                          077C  1826 ;
                          077C  1827 ;--
                          077C  1828
                          077C  1829 FIL$CHKFILHDR:
02     07 A1     91       077C  1830          CMPB    FH2$B_STRUCLEV(R1),#2   ;IS THIS STRUCTURE LEVEL 2?
          1E     12       0780  1831          BNEQ    30$                     ;BR IF NOT, REPORT ERROR
                          0782  1832 ;
                          0782  1833 ; STRUCTURE LEVEL 2
                          0782  1834 ;
7E     0C A1     3C       0782  1835 10$:     MOVZWL  FH2$W_FID_RVN(R1),-(SP) ;PUSH RELATIVE VOLUME NUMBER
7E     08 A1     D0       0786  1836          MOVL    FH2$W_FID_NUM(R1),-(SP) ;PUSH FILE ID ON STACK
          6E     B5       078A  1837 15$:     TSTW    (SP)                    ;FILE DELETED?
          18     13       078C  1838          BEQL    40$                     ;BRANCH IF YES
    8E    80     D1       078E  1839          CMPL    (R0)+,(SP)+             ;FILE NUM AND FILE SEQ NUM AGREE?
          0D     12       0791  1840          BNEQ    30$                     ;BRANCH IF NOT, BAD HEADER
          6E     D5       0793  1841          TSTL    (SP)                    ;CHECKING RVN?
          05     19       0795  1842          BLSS    20$                     ;BRANCH IF NOT
    6E    60     B1       0797  1843          CMPW    (R0),(SP)               ;RELATIVE VOLUME NUMBER AND
                          079A  1844                                          ;FILE NUMBER EXTENSION AGREE
          04     12       079A  1845          BNEQ    30$                     ;BRANCH IF NOT
          01     BA       079C  1846 20$:     POPR    #^M<R0>                 ;CLEAN OFF STACK
```

```
                    0C   11  079E   1847            BRB     FIL$CHECKSUM            ;GO VERIFY THE CHECKSUM
        50    0810 8F   3C  07A0   1848 30$:       MOVZWL  #SS$_BADFILEHDR,R0      ;THIS HEADER IS BAD
                         04  07A5   1849            RET
        50    0910 8F   3C  07A6   1850 40$:       MOVZWL  #SS$_NOSUCHFILE,R0      ;DELETED FILE
                         04  07AB   1851            RET
```

```
                   07AC  1853         .SBTTL  CHECKSUM - VALIDATE A CHECKSUM
                   07AC  1854 ;++
                   07AC  1855 ; FUNCTIONAL DESCRIPTION:
                   07AC  1856 ;
                   07AC  1857 ;       THIS ROUTINE CALCULATES AND CHECKS THE FILE11 CHECKSUM FOR
                   07AC  1858 ; FILE HEADERS AND THE HOMEBLOCK.
                   07AC  1859 ;
                   07AC  1860 ; CALLING SEQUENCE:
                   07AC  1861 ;
                   07AC  1862 ;       BSBW    FIL$CHECKSUM              ;CHECK FILE HEADER CHECKSUM
                   07AC  1863 ;       BSBW    FIL$CHECKSUM1             ;CHECK SPECIFIED NO. OF WORDS IN R0
                   07AC  1864 ;
                   07AC  1865 ; INPUT PARAMETERS:
                   07AC  1866 ;
                   07AC  1867 ;       R0 = NO. OF WORDS TO CHECK IF ENTERING AT CHECKSUM1
                   07AC  1868 ;       R1 = ADDRESS OF BUFFER TO CHECK
                   07AC  1869 ;
                   07AC  1870 ; IMPLICIT INPUTS:
                   07AC  1871 ;
                   07AC  1872 ;       NONE
                   07AC  1873 ;
                   07AC  1874 ; OUTPUT PARAMETERS:
                   07AC  1875 ;
                   07AC  1876 ;       RSB TO CALLER IF CHECKSUM IS OK
                   07AC  1877 ;       RET TO TOP LEVEL WITH ERROR CODE IN R0 IF CHECKSUM IS WRONG
                   07AC  1878 ;
                   07AC  1879 ; IMPLICIT OUTPUTS:
                   07AC  1880 ;
                   07AC  1881 ;       NONE
                   07AC  1882 ;
                   07AC  1883 ; COMPLETION CODES:
                   07AC  1884 ;
                   07AC  1885 ;       NONE
                   07AC  1886 ;
                   07AC  1887 ; SIDE EFFECTS:
                   07AC  1888 ;
                   07AC  1889 ;       NONE
                   07AC  1890 ;
                   07AC  1891 ;--
                   07AC  1892
                   07AC  1893 FIL$CHECKSUM:
50   00FF 8F   3C  07AC  1894         MOVZWL  #FH2$W_CHECKSUM@-1,R0   ;NO. OF WORDS TO CHECK
                   07B1  1895 FIL$CHECKSUM1:
          52   D4  07B1  1896         CLRL    R2                      ;INIT THE SUM
                   07B3  1897 10$:
     52   81   A0  07B3  1898         ADDW    (R1)+,R2                ;ACCUMULATE THE SUM
        FA 50  F5  07B6  1899         SOBGTR  R0,10$                  ;ONCE FOR EACH WORD
     61   52   B1  07B9  1900         CMPW    R2,(R1)                 ;CHECKSUM OK?
          01   12  07BC  1901         BNEQ    20$                     ;BRANCH IF NOT
               05  07BE  1902         RSB
                   07BF  1903 20$:
50   0808 8F   3C  07BF  1904         MOVZWL  #SS$_BADCHKSUM,R0       ;ERROR STATUS IN R0
               04  07C4  1905         RET
                   07C5  1906
                   07C5  1907
                   07C5  1908         .END
```

```
$$T1              = 00000001             FIL$A_DIR_FID       00000000
ARGCNT            = 00000000             FIL$A_DIR_OFID      0000001E
ASSIGN_DEV          00000213 R   02      FIL$A_IXFHDR        00000018
BADDIR              000004A4 R   02      FIL$B_DIR_LVL       00000012
BADDIR1             00000589 R   02      FIL$CACHE_INIT      00000180 RG  02
BADDIR2             00000418 R   02      FIL$CACHE_TRUNC     000001F4 RG  02
BADFILNAM           000004AA R   02      FIL$CHECKSUM        000007AC R   02
BADRET              000004A9 R   02      FIL$CHECKSUM1       000007B1 R   02
BADRET1             0000041B R   02      FIL$CHKFILHDR       0000077C R   02
BUFADR            = 0000000C             FIL$CVT_DTB         ********   X 02
CACHE_ADR         = 00000010             FIL$C_CACHE_ID    = 00000001
CACHE_SIZE        = 0000000C             FIL$C_DIR_SIZE      00000024  G
CHAN              = 00000004             FIL$C_SIZE          00000218  G
CHANADR           = 00000004             FIL$FINDFILID       00000358 RG  02
CMNSYS              00000000 R   02      FIL$GQ_CACHE        ********   X 02
DIR$B_NAMECOUNT   = 00000005             FIL$GT_DDDEV        ********   X 02
DIR$C_VERSION     = 00000008             FIL$GT_DDSTRING     ********   X 02
DIR$T_NAME        = 00000006             FIL$GT_TOPSYS       ********   X 02
DIR$W_FID         = 00000002             FIL$L_DIRMAX        0000000C
DIR$W_SIZE        = 00000000             FIL$L_DIRNXT        00000008
DIR$W_VERSION     = 00000000             FIL$L_DIROFF        00000004
DIR...            = FFFFFFFF             FIL$L_DIR_BFOFF     00000018
DIRBUF            = 00000010             FIL$L_DIR_LBN       00000014
DIRNAM              FFFFFFEA             FIL$L_LBNMAX        00000014
DIR_BFCNT           FFFFFFFC             FIL$L_LBNNXT        00000010
DIR_BUF             FFFFFFF8             FIL$L_LBNOFF        0000000C
DIR_CACHE_CNT     = 00000014             FIL$MOUNT           000002D3 RG  02
DIR_DESC            FFFFFFC8             FIL$OPENFILE        0000000A RG  02
DIR_LIST_SIZE     = 00000018             FIL$OPENFILE_1      00000013 RG  02
ENTRY               FFFFFFD0             FIL$Q_DIR_HDR       00000010
ENTRY_ADR           FFFFFFF4             FIL$RDCHKFILHDR     000005BA RG  02
EXIT_FILID_FND      00000555 R   02      FIL$RDWRTLBN        ********   X 02
FAT$B_RATTRIB     = 00000001             FIL$READVBN         00000685 RG  02
FAT$B_RTYPE       = 00000000             FIL$STATBLK         00000718 RG  02
FAT$C_VARIABLE    = 00000002             FIL$T_DIR_NAM       00000006
FAT$L_EFBLK       = 00000008             FIL$WRITEVBN        0000067E RG  02
FAT$M_NOSPAN      = 00000008             FIL$W_CACHE_ID      00000000
FAT$W_FFBYTE      = 0000000C             FIL$W_DIR_BFCNT     0000001C
FH2$B_MAP_INUSE   = 0000003A             FIL$W_DIR_BKCNT     00000010
FH2$B_MPOFFSET    = 00000001             FILDSC            = 00000008
FH2$B_STRUCLEV    = 00000007             FILE_NAME_DESC      FFFFFFC0
FH2$C_LEVEL2      = 00000200             FILHDR            = 00000004
FH2$L_FILECHAR    = 00000034             FILID             = 00000018
FH2$V_BIGFILNUM   = 0000000A             FILNAM            = 00000008
FH2$V_CONTIG      = 00000007             FIND_LEVEL2         000004B0 R   02
FH2$V_DIRECTORY   = 0000000D             FIND_LEVEL2_1       000004BC R   02
FH2$V_LEVEL2      = 00000009             FLAGS             = 00000020
FH2$W_CHECKSUM    = 000001FE             FM2$S_COUNT2      = 0000000E
FH2$W_EXT_FID     = 0000000E             FM2$S_FORMAT      = 00000002
FH2$W_FID_NUM     = 00000008             FM2$S_HIGHLBN     = 00000006
FH2$W_FID_RVN     = 0000000C             FM2$V_COUNT2      = 00000000
FH2$W_RECATTR     = 00000014             FM2$V_FORMAT      = 0000000E
FH2$W_STRUCLEV    = 00000006             FM2$V_HIGHLBN     = 00000008
FH2$W_VBNOFFSET   = 000001FE             FORMAT1             000006F8 R   02
FID                 FFFFFFFA             FORMAT2             0000070B R   02
FID$B_NMX         = 00000005             FORMAT3             000006E1 R   02
FID$C_MFD         = 00000004             FORMDIRSTRING       00000267 R   02
```

```
GETRTRVPTR                        000006D2 R      02          OP$_CVTFL                      = 0000004A
HDRCNT                            FFFFFFFC                    OP$_CVTFW                      = 00000049
HM2$B_STRUCLEV                  = 0000000D                    OP$_CVTGB                      = 000048FD
HM2$L_IBMAPLBN                  = 00000018                    OP$_CVTGF                      = 000033FD
HM2$L_MAXFILES                  = 0000001C                    OP$_CVTGH                      = 000056FD
HM2$W_CHECKSUM1                 = 0000003A                    OP$_CVTGL                      = 00004AFD
HM2$W_CLUSTER                   = 0000000E                    OP$_CVTGW                      = 000049FD
HM2$W_IBMAPSIZE                 = 00000020                    OP$_CVTHB                      = 000068FD
INCRSB                            00000708 R      02          OP$_CVTHD                      = 0000F7FD
INIRTRVPTRSCAN                    000006C1 R      02          OP$_CVTHF                      = 0000F6FD
IO$_READLBLK                    = 00000021                    OP$_CVTHG                      = 000076FD
IO$_WRITELBLK                   = 00000020                    OP$_CVTHL                      = 00006AFD
IOFUNCTION                      = FFFFFFFC                    OP$_CVTHW                      = 000069FD
IXFHDR                          = 0000000C                    OP$_CVTLD                      = 0000006E
LBN_CACHE_CNT                   = 00000018                    OP$_CVTLF                      = 0000004E
LOCAL_FLAGS                       FFFFFFD0                    OP$_CVTLG                      = 00004EFD
NAMBLK                            FFFFFFE0                    OP$_CVTLH                      = 00006EFD
NAMDSC                            FFFFFFD4                    OP$_CVTLP                      = 000000F9
OP$_ACBD                        = 0000006F                    OP$_CVTPL                      = 00000036
OP$_ACBF                        = 0000004F                    OP$_CVTPS                      = 00000008
OP$_ACBG                        = 00004FFD                    OP$_CVTPT                      = 00000024
OP$_ACBH                        = 00006FFD                    OP$_CVTRDL                     = 0000006B
OP$_ADDD2                       = 00000060                    OP$_CVTRFL                     = 0000004B
OP$_ADDD3                       = 00000061                    OP$_CVTRGL                     = 00004BFD
OP$_ADDF2                       = 00000040                    OP$_CVTRHL                     = 00006BFD
OP$_ADDF3                       = 00000041                    OP$_CVTSP                      = 00000009
OP$_ADDG2                       = 000040FD                    OP$_CVTTP                      = 00000026
OP$_ADDG3                       = 000041FD                    OP$_CVTWD                      = 0000006D
OP$_ADDH2                       = 000060FD                    OP$_CVTWF                      = 0000004D
OP$_ADDH3                       = 000061FD                    OP$_CVTWG                      = 00004DFD
OP$_ADDP4                       = 00000020                    OP$_CVTWH                      = 00006DFD
OP$_ADDP6                       = 00000021                    OP$_DIVD2                      = 00000066
OP$_ASHP                        = 000000F8                    OP$_DIVD3                      = 00000067
OP$_CLRD                        = 0000007C                    OP$_DIVF2                      = 00000046
OP$_CLRF                        = 000000D4                    OP$_DIVF3                      = 00000047
OP$_CLRG                        = 0000007C                    OP$_DIVG2                      = 000046FD
OP$_CLRH                        = 00007CFD                    OP$_DIVG3                      = 000047FD
OP$_CMPD                        = 00000071                    OP$_DIVH2                      = 000066FD
OP$_CMPF                        = 00000051                    OP$_DIVH3                      = 000067FD
OP$_CMPG                        = 000051FD                    OP$_DIVP                       = 00000027
OP$_CMPH                        = 000071FD                    OP$_EDITPC                     = 00000038
OP$_CMPP3                       = 00000035                    OP$_EMODD                      = 00000074
OP$_CMPP4                       = 00000037                    OP$_EMODF                      = 00000054
OP$_CRC                         = 0000000B                    OP$_EMODG                      = 000054FD
OP$_CVTBD                       = 0000006C                    OP$_EMODH                      = 000074FD
OP$_CVTBF                       = 0000004C                    OP$_MATCHC                     = 00000039
OP$_CVTBG                       = 00004CFD                    OP$_MNEGD                      = 00000072
OP$_CVTBH                       = 00006CFD                    OP$_MNEGF                      = 00000052
OP$_CVTDB                       = 00000068                    OP$_MNEGG                      = 000052FD
OP$_CVTDF                       = 00000076                    OP$_MNEGH                      = 000072FD
OP$_CVTDH                       = 000032FD                    OP$_MOVD                       = 00000070
OP$_CVTDL                       = 0000006A                    OP$_MOVF                       = 00000050
OP$_CVTDW                       = 00000069                    OP$_MOVG                       = 000050FD
OP$_CVTFB                       = 00000048                    OP$_MOVH                       = 000070FD
OP$_CVTFD                       = 00000056                    OP$_MOVP                       = 00000034
OP$_CVTFG                       = 000099FD                    OP$_MOVTC                      = 0000002E
OP$_CVTFH                       = 000098FD                    OP$_MOVTUC                     = 0000002F
```

```
OP$_MULD2                        = 00000064                TMPRTRVLEN                       FFFFFFF8
OP$_MULD3                        = 00000065                VBN                            = 00000008
OP$_MULF2                        = 00000044
OP$_MULF3                        = 00000045
OP$_MULG2                        = 000044FD
OP$_MULG3                        = 000045FD
OP$_MULH2                        = 000064FD
OP$_MULH3                        = 000065FD
OP$_MULP                         = 00000025
OP$_POLYD                        = 00000075
OP$_POLYF                        = 00000055
OP$_POLYG                        = 000055FD
OP$_POLYH                        = 000075FD
OP$_SCANC                        = 0000002A
OP$_SKPC                         = 0000003B
OP$_SPANC                        = 0000002B
OP$_SUBD2                        = 00000062
OP$_SUBD3                        = 00000063
OP$_SUBF2                        = 00000042
OP$_SUBF3                        = 00000043
OP$_SUBG2                        = 000042FD
OP$_SUBG3                        = 000043FD
OP$_SUBH2                        = 000062FD
OP$_SUBH3                        = 000063FD
OP$_SUBP4                        = 00000022
OP$_SUBP6                        = 00000023
OP$_TSTD                         = 00000073
OP$_TSTF                         = 00000053
OP$_TSTG                         = 000053FD
OP$_TSTH                         = 000073FD
OPENFILE_2                         00000015 R      02
PLACEMENT                          000006EF R      02
PREFIX_DIR_LIST                    FFFFFFA8
PSL$S_CURMOD                     = 00000002
PSL$V_CURMOD                     = 00000018
RDWRTVBN                           0000068A R      02
READ_DIR_HEADER                    0000041C R      02
READ_DIR_LBN                       0000058C R      02
RTRVPTRBUF                       = 00000010
RTRVPTRLEN                       = 0000000C
SAVABS...                        = FFFFFFF0
SCRATCHSIZE                        FFFFFFA8
SCRATCH_SIZE                       FFFFFFD0
SS$_BADCHKSUM                    = 00000808
SS$_BADFILEHDR                   = 00000810
SS$_BADFILENAME                  = 00000818
SS$_BADIRECTORY                  = 00000828
SS$_ENDOFFILE                    = 00000870
SS$_FILESTRUCT                   = 000008C0
SS$_NOPRIV                       = 00000024
SS$_NORMAL                       = 00000001
SS$_NOSUCHFILE                   = 00000910
STATBLK                          = 00000008
STORE3DIGITS                       0000024E R      02
SYS$ASSIGN                         ********   GX   02
SYS$CMKRNL                         ********   GX   02
TMPRTRVDSC                         FFFFFFF0
```

```
                                               +----------------+
                                               ! Psect synopsis !
                                               +----------------+

PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (     0.)  00 (  0.)  NOPIC    USR    CON    ABS    LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                           FFFFFFFC  (     0.)  01 (  1.)  NOPIC    USR    CON    ABS    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
YFILEREAD                       000007C5  ( 1989.)  02 (  2.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD     WRT NOVEC BYTE

                                            +------------------------+
                                            ! Performance indicators !
                                            +------------------------+

Phase                     Page faults    CPU Time       Elapsed Time
-----                     -----------    --------       ------------
Initialization                     31    00:00:00.05    00:00:01.87
Command processing                107    00:00:00.56    00:00:06.37
Pass 1                            630    00:00:26.32    00:01:19.14
Symbol table sort                   0    00:00:02.50    00:00:08.84
Pass 2                            394    00:00:07.39    00:00:27.05
Symbol table output                36    00:00:00.25    00:00:00.51
Psect synopsis output               1    00:00:00.04    00:00:00.24
Cross-reference output              0    00:00:00.00    00:00:00.00
Assembler run totals             1201    00:00:37.11    00:02:04.02

The working set limit was 1950 pages.
126068 bytes (247 pages) of virtual memory were used to buffer the intermediate code.
There were 90 pages of symbol table space allocated to hold 1556 non-local and 96 local symbols.
4660 source lines were read in Pass 1, producing 18 object records in Pass 2.
154 pages of virtual memory were used to define 152 macros.

                                           +--------------------------+
                                           ! Macro library statistics !
                                           +--------------------------+

Macro library name                          Macros defined
------------------                          --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                           6
_$255$DUA28:[SYSLIB]STARLET.MLB;2                       15
TOTALS (all libraries)                                  21

1638 GETS were required to define 21 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:FILEREAD/OBJ=OBJ$:FILEREAD MASD$:[EMULAT.SRC]MISSING/UPDATE=(MASD$:[EMULAT.ENH]MISSING)+MASD$:[SYS.SRC]FILEREAD/UPDAT
```

0374 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY